Haikal Hilmi

Data and Software Engineer specializing in resilient data pipelines, large-scale scraping, cloud systems, HPC infrastructure, and full-stack products. Backend Engineer intern at ByteDance working on systems handling 100,000+ requests per second, a 5.0-star Upwork freelancer, and a multiple international hackathon winner.

This document is optimized for search, retrieval-augmented generation (RAG), and AI knowledge bases. It is generated from the live portfolio data source.

Contact and profiles

- Website: https://haikalhilmi.my.id
- LinkedIn: https://www.linkedin.com/in/haikalhilmi/
- Upwork: https://www.upwork.com/freelancers/~01ff216c8a0f8a68a1
- Project inquiry: https://tally.so/r/81VNPz

Core capabilities

- Data engineering: scraping, ingestion, ETL, orchestration, queues, search, observability, and production reliability.
- Software engineering: full-stack web products, APIs, microservices, integrations, authentication, and scalable architecture.
- Infrastructure: on-premise HPC, Docker workloads, monitoring, capacity planning, and cloud deployment.
- Applied AI: LLM products, sentiment analysis, speech-to-text, retrieval, automation, and data enrichment.

Selected scale and infrastructure

- 8 HPC Servers - 4 main + 4 additional
- 64 Cores (max) - AMD EPYC 7502P
- 200GB+ RAM (max) - Ubuntu 22.04 / CentOS
- 100K+ Records / day - Pipeline throughput
- 4x Tesla V100 GPUs - NVIDIA DGX Station
- 128GB VRAM (total) - 32GB per GPU

Professional experience

ByteDance

- Role: Backend Engineer Intern
- Period: Feb 2025 - Jun 2025
- Location: Yogyakarta, ID
- Impact: Built a microservice handling 100,000+ requests/second.
- Context: Contributed to Tokopedia x TikTok-Shop system integrations.
- Technologies: Go, gRPC, RocketMQ, NSQ, Bytecloud
- Highlights:
- Contributed to Tokopedia x TikTok-Shop system integrations.
- Added Lark support to the internal dev kit, cutting log spam ~60%.

Engineering Research & Innovation Center, FT UGM

- Role: Full Stack Software Engineer

- Period: Jul 2025 - Jan 2026
- Location: Yogyakarta, ID
- Impact: Ran UGM Career Summit for 2,800+ concurrent users with zero downtime.
- Technologies: Next.js, React, TypeScript, Supabase
- Highlights:
- Engineered a white-label ticketing architecture with QR validation under 2 seconds.
- Built an automated document-generation pipeline for an international symposium.

Semesta Data Digital

- Role: Data Engineer Contract
- Period: Jan 2024 - Nov 2025
- Location: Jakarta, ID
- Impact: Designed HPC data pipelines - reliability +25%, processing time 40%.
- Context: Operated an 8-server HPC cluster ingesting social-media & news data at scale.
- Technologies: Python, Elasticsearch, PostgreSQL, RabbitMQ, Docker, Grafana
- Highlights:
- Architected a resilient social-media data pipeline ingesting from diverse sources.
- Built Grafana + Prometheus monitoring with alerts that prevented server crashes.

Tilikan Indonesia

- Role: Web App Developer Contract
- Period: Sep 2023 - Mar 2024
- Location: Yogyakarta, ID
- Impact: Analytics dashboards - reliability +50%, cross-team efficiency +70%.
- Technologies: Next.js 14, React, TypeScript, MongoDB
- Highlights:
- Integrated multiple APIs to streamline interdepartmental workflows.
- Shipped a document-conversion feature for better document management.

Global Data Inspirasi

- Role: Web App Developer Contract
- Period: Jan 2023 - Jan 2024
- Location: Yogyakarta, ID
- Impact: Led a team building real-time data dashboards and reusable components.
- Technologies: Next.js 13, React, TypeScript, TanStack
- Highlights:
- Applied Agile to accelerate delivery and raise software quality.
- Delivered scalable apps that gave actionable business insights.

Telkom Indonesia

- Role: Datacomm Product Dev Intern
- Period: Jan 2024 - Mar 2024
- Location: Jakarta, ID
- Impact: Penetration-tested critical web platforms and hardened their security.
- Technologies: Python, SD-WAN, Tableau
- Highlights:
- Reported vulnerabilities and coordinated fixes with vendors.
- Aligned security measures with business objectives in an Agile team.

Software engineering projects

PwC Securing AI

- ID: pwc
- Category: Software Engineering
- Context: competition
- Role: Lead Engineer full-stack (team of 4)
- Award: 1st in Indonesia 2nd in Southeast Asia
- Summary: An AI assistant that recommends the best credit options for each user based on their income and spending.
- Problem: People struggle to pick the right credit product for their finances.
- Solution: Built a GPT-style chatbot + dashboard in 14 days, matching users to real credit products.
- Result: From concept to a working AI product in just 14 days.
- Technologies: Next.js 13, Azure, MongoDB 7, Node.js, OpenAI
- Cover image: https://haikalhilmi.my.id/Porto/lensightoripwc.png
- Key decisions:
- MongoDB for fast iteration on uncertain, evolving metrics under a 14-day deadline.
- Ran the SDLC in parallel with the UI/UX team - execute first, polish last - to ship in 14 days.
- Azure (hackathon sponsor) + Next.js as the team's fastest, most familiar path to delivery.
- Portfolio detail: https://haikalhilmi.my.id/projects/pwc

GMAT Satellite

- ID: gmat
- Category: Software Engineering
- Context: competition
- Role: Technical Lead & Project Manager
- Award: 3rd place Teknofest Turkey 2022 (international)
- Summary: Real-time ground control for an international satellite competition-telemetry, mapping, and live 3D visualization.
- Problem: Needed real-time monitoring of satellite telemetry.
- Solution: Telemetry dashboard, mapping, and 3D satellite visualization.
- Result: Delivered live telemetry under real competition conditions.
- Technologies: React, Three.js, Raspberry Pi, Arduino, XBEE
- Cover image: https://haikalhilmi.my.id/home/portfolio/gmat.png
- Portfolio detail: https://haikalhilmi.my.id/projects/gmat

University Onboarding Platform

- ID: ppsmb
- Category: Software Engineering
- Context: client
- Role: IT Team Lead (team of 7)
- Summary: Onboarding website and staff task-management system for a major university's incoming students.
- Problem: Committee coordination and information delivery weren't centralized.
- Solution: Information website plus a task management application.
- Result: 1.3M visitors in the first month, with 300+ staff coordinated in one place.
- Technologies: Next.js 12, TailwindCSS, Strapi
- Cover image: https://haikalhilmi.my.id/home/portfolio/ppsmb.png
- Portfolio detail: https://haikalhilmi.my.id/projects/ppsmb

Robota - AI Hotel Platform

- ID: robota-new
- Category: Software Engineering
- Context: client
- Role: Frontend Lead
- Summary: A B2B platform that tracks hotel room prices across OTAs (which shift hourly) and analyzes competitor pricing & availability.
- Problem: Hotels' OTA prices shift hour to hour, and they couldn't easily see how competitors compared.
- Solution: Migrated the frontend to Next.js 13 (SSR/SSG) and built dashboards over a custom data backend with AI competitor analysis.
- Result: A faster, SEO-ready rebuild, migrated off a legacy React SPA.
- Technologies: Next.js 13, React, TypeScript, TailwindCSS, TanStack Query
- Cover image: https://haikalhilmi.my.id/home/portfolio/robotav2.png
- Key decisions:
- Migrated off a legacy React SPA to Next.js 13 (SSR/SSG) to fix SEO and slow dashboard loads.
- TanStack Query caches each backend request, keeping heavy real-time dashboards responsive.
- Set up the frontend architecture and folder structure for the team.
- Portfolio detail: https://haikalhilmi.my.id/projects/robota-new

Tilikan

- ID: tilikan
- Category: Software Engineering
- Context: client
- Role: Web App Developer
- Summary: Interactive analytics and data-visualization dashboards built for client reporting.
- Problem: Clients needed easy-to-understand visualizations that integrate with their systems.
- Solution: Interactive dashboards, geographic maps, and PDF export.
- Result: Boosted system reliability ~50% and cross-team efficiency ~70%.
- Technologies: Next.js 14, TanStack, TailwindCSS, ECharts, ChartJS
- Cover image: https://haikalhilmi.my.id/home/portfolio/tilikan.png
- Portfolio detail: https://haikalhilmi.my.id/projects/tilikan

Petadata

- ID: petadata
- Category: Software Engineering
- Context: client
- Summary: A monitoring and analytics platform that brings scattered public data into one place.
- Problem: Data was scattered across many sources, making centralized analysis hard.
- Solution: Data visualization and aggregation dashboard.
- Result: Turns fragmented data into one clear view for faster decisions.
- Technologies: React, Next.js, ChartJS, REST API
- Cover image: https://haikalhilmi.my.id/home/portfolio/petadata.png
- Portfolio detail: https://haikalhilmi.my.id/projects/petadata

Atensi - Sentiment Monitoring

- ID: atensi
- Category: Software Engineering
- Context: client
- Summary: Real-time public sentiment and issue monitoring across social media and news.

- Problem: Hard to know which topics are gaining public attention in real time.
- Solution: Visualizes trends, sentiment, and trending topics.
- Result: Spot trending issues in real time-before they escalate.
- Technologies: Next.js, ChartJS, Elasticsearch
- Cover image: https://haikalhilmi.my.id/home/portfolio/atensi.png
- Portfolio detail: https://haikalhilmi.my.id/projects/atensi

Personal Safety App

- ID: ganjar
- Category: Software Engineering
- Context: competition
- Award: 3rd place national hackathon
- Summary: An award-winning, location-based personal-safety app with live maps.
- Problem: Hard to report and monitor unsafe areas.
- Solution: Mobile app with Google Maps integration.
- Result: Helps people report and avoid unsafe areas, powered by live maps.
- Technologies: Flutter, Google Maps API
- Cover image: https://haikalhilmi.my.id/home/portfolio/ganjar.png
- Portfolio detail: https://haikalhilmi.my.id/projects/ganjar

FoodCycle

- ID: foodcycle
- Category: Software Engineering
- Context: competition
- Role: Team Lead
- Award: 1st place Hackbiz 180DC UGM 2022
- Summary: An ML platform that detects food waste via image recognition and connects waste generators with distributors.
- Problem: Edible surplus food goes to waste instead of reaching those who can use it.
- Solution: Image recognition to identify food, plus a marketplace linking generators to distributors.
- Technologies: Machine Learning, Image Recognition, Web
- Cover image: https://haikalhilmi.my.id/Porto/foodcycle.png
- Portfolio detail: https://haikalhilmi.my.id/projects/foodcycle

Omnisocials

- ID: omnisocials
- Category: Software Engineering
- Context: client
- Role: Full Stack Engineer (freelance)
- Status: In production
- Summary: A SaaS platform to manage and schedule content across many social networks from one dashboard.
- Problem: Teams had to juggle many platforms to publish and monitor their content.
- Solution: A unified dashboard that connects to major social networks via their official APIs.
- Result: Manage all your social networks from a single dashboard.
- Technologies: Next.js, Node.js, PostgreSQL, TailwindCSS
- Cover image: https://haikalhilmi.my.id/Porto/omnisocial.png
- Key decisions:
- Cross-platform authentication - secure, scalable OAuth across 10+ social APIs with robust token-lifecycle management for stable connections.

- Scalable job scheduling - a background processing system for high-volume scheduled publishing with predictable resource usage.
- Real-time feedback - live client-server updates so users see task status instantly.
- API traffic management - centralized request handling and throttling to respect diverse third-party rate limits without losing throughput.
- Portfolio detail: https://haikalhilmi.my.id/projects/omnisocials

Create Carousels

- ID: create-carousels
- Category: Software Engineering
- Context: client
- Role: Full Stack Engineer
- Summary: An AI tool that turns articles, ideas, or documents into ready-to-post social carousels.
- Problem: Making carousels is time-consuming: research, copywriting, and per-slide design.
- Solution: Turns text input into post-ready carousels with AI & design templates.
- Result: Turns hours of content design into minutes.
- Technologies: Next.js, TypeScript, OpenAI API, TailwindCSS, Supabase
- Cover image: https://haikalhilmi.my.id/Porto/create-carousels.png
- Key decisions:
- AI content generation - turns articles, ideas, or documents into structured, slide-ready copy with prompt design for consistent tone and format.
- Template-driven design - programmatic slide layouts so generated content maps to on-brand visuals automatically.
- Export pipeline - renders finished carousels into post-ready images for direct publishing.
- Portfolio detail: https://haikalhilmi.my.id/projects/create-carousels

Festify

- ID: festify
- Category: Software Engineering
- Context: client
- Role: Full Stack Engineer
- Summary: A white-label event ticketing platform with built-in QR validation.
- Problem: Event organizers needed a reusable ticketing system for various events.
- Solution: Reusable ticketing architecture with QR validation.
- Result: Reusable ticketing that powers multiple live events-no rebuilds.
- Technologies: Next.js, Supabase, TypeScript
- Cover image: https://haikalhilmi.my.id/home/portfolio/festify.png
- Portfolio detail: https://haikalhilmi.my.id/projects/festify

Online Competition Platform

- ID: nesco
- Category: Software Engineering
- Context: client
- Role: Frontend Lead (team of 6)
- Summary: Website and dashboards that run an online competition end-to-end.
- Problem: Needed a responsive competition system and dashboard.
- Solution: Website, authentication, and admin/user dashboards.
- Result: Ran a full competition for participants and organizers alike.
- Technologies: Next.js 12, TailwindCSS, REST API, Node.js
- Cover image: https://haikalhilmi.my.id/home/portfolio/nesco.png
- Portfolio detail: https://haikalhilmi.my.id/projects/nesco

Linkyin.bio

- ID: linkynbio
- Category: Software Engineering
- Context: client
- Role: Full Stack Engineer
- Summary: A bio-link platform (like Linktree) for creators and businesses.
- Problem: Creators & businesses share many different links, making it hard for audiences.
- Solution: Manage multiple links, set priority, and toggle them-all in one URL.
- Result: One link that holds a user's entire online presence.
- Technologies: Next.js, TypeScript, TailwindCSS, Supabase
- Cover image: https://haikalhilmi.my.id/Porto/linkyinbio.jpeg
- Key decisions:
- Customizable link pages - manage, reorder, and toggle multiple links behind one shareable URL.
- Fast, SEO-friendly delivery - server-rendered profile pages optimized for quick load.
- Auth & data - user accounts and saved profiles.
- Portfolio detail: https://haikalhilmi.my.id/projects/linkynbio

Robota - First Generation

- ID: robota-old
- Category: Software Engineering
- Context: client
- Role: Web App Developer
- Summary: The first generation of the Robota platform, built on React and WebSockets.
- Problem: The dashboard handled large data and rendering poorly.
- Solution: React + WebSocket analytics dashboard.
- Result: The proven foundation the current product was built on.
- Technologies: React, TypeScript, WebSocket, ChartJS
- Cover image: https://haikalhilmi.my.id/home/portfolio/robotav1.png
- Portfolio detail: https://haikalhilmi.my.id/projects/robota-old

Blockchain Supply Chain

- ID: blockchain
- Category: Software Engineering
- Summary: A blockchain app for end-to-end supply-chain traceability.
- Problem: Supply-chain oversight from farming through to sale.
- Solution: Website with smart contracts on Solana.
- Result: Traceability from farm to sale, secured on-chain with smart contracts.
- Technologies: Next.js 13, Node.js, Solana
- Cover image: https://haikalhilmi.my.id/home/portfolio/agrichain.png
- Portfolio detail: https://haikalhilmi.my.id/projects/blockchain

Kembangkan Software Studio

- ID: kembangkan
- Category: Software Engineering
- Context: client
- Summary: A project-based software studio supporting small businesses.
- Solution: AI services, web development, and UI/UX.
- Result: Led a team that shipped 8+ products, from websites to ticketing systems.
- Technologies: AI, Web Development, UI/UX

- Cover image: https://haikalhilmi.my.id/Porto/Kembangkan.mp4
- Portfolio detail: https://haikalhilmi.my.id/projects/kembangkan

Data engineering projects

Android Ad Evidence Capture & DSP Attribution

- ID: android-ad-evidence
- Category: Data Engineering
- Context: client
- Role: Automation Engineer / Python Engineer
- Status: Android advertising verification PoC
- Scale: 463 flows 10 ad networks 7-minute lab session
- Summary: An automated Android ad-inspection system that captures creatives, UI evidence, click destinations, and network traffic to verify delivery paths and investigate DSP attribution such as Smadex.
- Problem: A screenshot cannot prove which buying platform delivered an ad, while dynamic creatives, stale intents, certificate restrictions, freezes, and non-rooted devices make reliable evidence collection difficult.
- Solution: Built a dual-path pipeline: ADB and uiautomator2 record what users see and safely inspect interactions, while mitmproxy captures delivery-chain signals for rule-based correlation and tamper-evident archiving.
- Result: Tamper-evident evidence bundles 95.69% coverage across 90 tests.
- Technologies: Python 3.11, uiautomator2, ADB, mitmproxy, Pillow, ImageHash, PowerShell, Pytest, mypy, JSON Lines, SHA-256
- Source & evidence: https://github.com/Harmerz/Android-advertising-verification-PoC
- Pipeline: Launch: PowerShell + Python CLI -> Control & detect: ADB + uiautomator2 -> Capture evidence: UI + screenshots + safe clicks -> Capture network: ADB reverse + mitmproxy -> Attribute & archive: Rules + JSONL + SHA-256
- Key decisions:
- Kept UI and network capture independent: Android automation proves what the user saw, while intercepted flows provide delivery-chain evidence.
- Required explicit Ad Info, redirect, client-signature, or reporting signals for confirmed DSP attribution; creatives alone remain candidates, never proof.
- Captured broadly, matched later, and recorded creative URLs without fetching them-preserving limited-window evidence without creating accidental impressions or clicks.
- Externalized provider indicators in JSON and separated attribution into confirmed, candidate, and notdetected states.
- Protected artifacts with append-only SHA-256 manifests, recorded measured geo metadata, and added ANR recovery, cold restarts, and proxy restoration for long sessions.
- Portfolio detail: https://haikalhilmi.my.id/projects/android-ad-evidence

Competitor SEO Monitoring Pipeline

- ID: competitor-seo-monitoring
- Category: Data Engineering
- Context: owned
- Role: Data / Backend Engineer
- Status: Portfolio project Local-first PoC
- Scale: 40 live fetches 25 complete articles 0 failures
- Summary: A local-first pipeline for monitoring competitor content and SEO changes, using Daypass as the primary website and ResortPass as the competitor.
- Problem: Recurring competitor SEO reviews are slow and hard to audit when assembled manually, while raw HTML comparisons create noisy alerts from navigation, timestamps, and dynamic widgets.

- Solution: Built two isolated paths: a free crawler, snapshot, diff, dashboard, and workbook workflow; plus optional DataForSEO enrichment guarded by dry runs, environment-only credentials, caching, and a hard cost ceiling.
- Result: Complete article records paid enrichment capped below $1.
- Technologies: Python 3.11+, JSONL / JSON, DataForSEO API v3, Node.js, @oai/artifact-tool, HTML, CSS, JavaScript, unittest
- Source & evidence: https://github.com/Harmerz/Competitor-SEO-Monitoring-Pipeline
- Pipeline: Discover: robots.txt + sitemaps + links -> Crawl: Bounded Python crawler -> Extract: Metadata + normalized content -> Store & compare: Immutable JSONL + diffs -> Enrich: Approved DataForSEO requests -> Report: Dashboard + auditable XLSX
- Key decisions:
- Used standard-library Python for the core crawler so the collection path stays reproducible without third-party Python dependencies.
- Chose robots-aware sequential crawling for ethical, auditable rate control instead of maximum throughput.
- Compared normalized main text for meaningful changes; raw HTML hashes remain diagnostic signals only.
- Made dated JSONL snapshots immutable unless replacement is explicitly requested with --force.
- Put DataForSEO behind a trust boundary: dry-run by default, environment-only credentials, allowlisted hosts, immutable caching, and a cost check before every request.
- Never fabricates missing SEO metrics; workbook formulas reference raw-data sheets and the dashboard binds only to loopback.
- Portfolio detail: https://haikalhilmi.my.id/projects/competitor-seo-monitoring

Cross-Platform Customer Review Scraper

- ID: cross-platform-review-scraper
- Category: Data Engineering
- Context: owned
- Role: Data / Backend Engineer
- Status: Portfolio project Cross-platform review extraction
- Scale: 3 review platforms 25 sample records
- Summary: A Python scraping toolkit that collects public customer reviews and app metadata from Trustpilot, Google Play, and the Apple App Store for customer research and sentiment analysis.
- Problem: Review research is fragmented across platforms with different delivery models: dynamic Next.js data on Trustpilot, paginated tokens on Google Play, and limited page-rendered public review data on the App Store.
- Solution: Built isolated scrapers for each source, then normalized review text, ratings, authors, dates, metadata, and source URLs into analysis-friendly JSON with deterministic deduplication and committed samples.
- Result: Normalized review JSON source URLs preserved.
- Technologies: Python 3.11+, Patchright, google-play-scraper, Apple iTunes Lookup API, App Store public web JSON, JSON, PowerShell, Google-style docstrings
- Source & evidence: https://github.com/Harmerz/Cross-Platform-Customer-Review-Scraper
- Pipeline: Discover: Profile URLs + package/app IDs -> Fetch: Patchright + platform endpoints -> Extract: Reviews + app/business metadata -> Normalize: Analysis-friendly JSON records -> Deduplicate: Review IDs + fallback keys -> Store: JSON outputs + trimmed samples
- Key decisions:
- Kept one script per platform so source-specific fetching, pagination, and fallback behavior remain isolated and maintainable.
- Used a Patchright browser context for Trustpilot before reading public Next.js JSON, reusing a realistic session without scraping visual markup.
- Prioritized App Store page-rendered JSON and retained the legacy RSS path only as a fallback because the public RSS feed is often empty.
- Preserved source URLs, deterministic review IDs, and fallback text/date keys so records remain auditable and deduplicated across runs.
- Stored trimmed portfolio samples separately from full generated outputs and documented platform limitations rather than hiding or estimating missing coverage.

- Portfolio detail: https://haikalhilmi.my.id/projects/cross-platform-review-scraper

Social Media Data Pipeline

- ID: pipeline
- Category: Data Engineering
- Context: client
- Role: Data Engineer
- Scale: Hundreds of thousands of records / day
- Summary: The pipeline behind the social-media & news scrapers - powering two-sided sentiment analysis and NER for a public-sector communications team.
- Problem: Data volume is enormous and must be processed continuously.
- Solution: HPC pipeline with monitoring and a queue system.
- Result: Cut processing time ~40% and improved reliability ~25%, monitored 24/7.
- Technologies: RabbitMQ, Docker, Elasticsearch, Grafana, Prometheus, Airflow
- Pipeline: Sources: Scrapers -> Queue: RabbitMQ -> Process: Docker workers -> Store: Elasticsearch -> Monitor: Grafana + Prometheus
- Key decisions:
- RabbitMQ for flexible routing - speed from scrape to display wasn't the bottleneck, so simple routing beat Kafka's complexity.
- Elasticsearch for fast full-text & context search across large volumes of news and social text.
- An alert bot pings the dev team the moment a scraper returns no data - kept uptime ~99% with fast recovery.
- On-prem HPC: the ~40% efficiency gain freed capacity for co-located AI and queue workloads.
- Portfolio detail: https://haikalhilmi.my.id/projects/pipeline

Instagram Scraping

- ID: instagram
- Category: Data Engineering
- Context: client
- Role: Data Engineer (solo)
- Scale: Up to ~50K posts & ~1M comments / day
- Summary: Large-scale Instagram data collection for AI sentiment analysis and competitor monitoring.
- Problem: Hard to obtain competitor engagement data automatically.
- Solution: Mass scraping of Instagram posts & comments.
- Result: Clean, structured engagement data-delivered daily.
- Technologies: Python, Selenium, PostgreSQL, Airflow
- Pipeline: Source: Instagram -> Crawl: Selenium -> Orchestrate: Airflow -> Store: PostgreSQL -> Analyze: Sentiment AI
- Key decisions:
- Benchmarked CPU, RAM, disk, and network per scraper - part of my undergraduate thesis research.
- Portfolio detail: https://haikalhilmi.my.id/projects/instagram

Twitter / X Scraping

- ID: twitter
- Category: Data Engineering
- Context: client
- Role: Data Engineer (solo)
- Scale: Up to ~75K tweets / day
- Summary: Distributed Twitter/X scraping for AI sentiment analysis and public-conversation monitoring.
- Problem: Client needed real-time monitoring of public conversations.
- Solution: A distributed Twitter crawler with a queue system.

- Result: A real-time pulse on public conversation.
- Technologies: Python, Selenium, RabbitMQ, Elasticsearch
- Pipeline: Source: Twitter / X -> Crawl: Selenium -> Queue: RabbitMQ -> Store: Elasticsearch -> Analyze: Sentiment AI
- Portfolio detail: https://haikalhilmi.my.id/projects/twitter

YouTube Scraping

- ID: youtube
- Category: Data Engineering
- Context: client
- Role: Data Engineer
- Scale: Up to ~5K videos & ~500K comments / day
- Summary: YouTube video and comment intelligence with automatic transcription.
- Problem: Needed large-scale analysis of videos and comments.
- Solution: Video & comment crawling pipeline + automated transcript analysis.
- Result: Searchable video insights, transcripts included.
- Technologies: Python, Elasticsearch, RabbitMQ, YouTube API
- Pipeline: Source: YouTube API -> Crawl: Python -> Queue: RabbitMQ -> Transcribe: Speech-to-Text -> Store: Elasticsearch
- Portfolio detail: https://haikalhilmi.my.id/projects/youtube

News Scraping

- ID: news
- Category: Data Engineering
- Context: client
- Role: Data Engineer
- Scale: 226K+ articles / week
- Summary: A unified crawler across ~2,900 news portals - automatic extraction for most via newspaper4k, with a custom-parser fallback for ~100 sources it couldn't handle out of the box.
- Problem: newspaper4k covers most portals with just a URL, but ~100 sources have markup it can't parse correctly.
- Solution: Built a reusable custom-parser template, adapted per portal with small tweaks instead of one-off scripts for each.
- Result: One consistent, clean feed across ~2,900 portals, with a maintainable fallback layer for the long tail.
- Technologies: Python, BeautifulSoup, Selenium, newspaper4k
- Pipeline: Sources: News portals -> Crawl: BeautifulSoup / newspaper4k -> Extract: Content parser -> Store: Database -> Analyze: NER + Sentiment AI
- Portfolio detail: https://haikalhilmi.my.id/projects/news

Lead Generation

- ID: leadgen
- Category: Data Engineering
- Context: client
- Role: Data Engineer
- Scale: Thousands of leads
- Summary: Automated B2B and B2C prospecting-email and company data on demand.
- Problem: Finding business prospects manually is hard.
- Solution: Automated email & company-data discovery.
- Result: Thousands of qualified prospects, sourced automatically.
- Technologies: Python, Automation, Apollo, ZoomInfo
- Pipeline: Sources: Apollo / ZoomInfo -> Automate: Python -> Enrich: Dedup + verify -> Deliver: CSV / CRM

- Portfolio detail: https://haikalhilmi.my.id/projects/leadgen

In-Video Word Finder

- ID: kick
- Category: Data Engineering
- Context: client
- Role: Sole Engineer
- Scale: 100+ hours of video processed
- Summary: Finds the exact moment a streamer says a target word (e.g. "LFG") for marketing clips - speech to searchable text via AI.
- Problem: A marketing client needed to catch specific words spoken across long Kick.com streams.
- Solution: Transcribe with Whisper, index words with timestamps, then auto-cut the clip at that moment.
- Result: Delivers the exact clip + URL where each searched word is spoken.
- Technologies: Python, Whisper, GPU
- Pipeline: Source: Kick.com -> Extract: Video -> Audio -> Transcribe: Speech-to-Text AI -> Index: Searchable text -> Output: Word finder
- Key decisions:
- Ran Whisper (medium) locally on a laptop GPU - accurate enough for keyword search at zero STT API cost.
- Batch-processed 100+ hours of video; indexed transcripts with timestamps to auto-cut clips.
- Whisper's language detection handled mixed-language streams.
- Portfolio detail: https://haikalhilmi.my.id/projects/kick

Auto-Engagement Bot

- ID: twitter-bot
- Category: Data Engineering
- Context: client
- Role: Data Engineer
- Scale: Hundreds of comments / day
- Summary: Automated, on-topic commenting built on the Twitter/X scraping system.
- Solution: Scheduled automated commenting on relevant conversations.
- Result: Hands-free engagement at scale.
- Technologies: Python, Automation
- Pipeline: Source: Twitter / X -> Filter: Relevance match -> Schedule: Python -> Act: Auto-comment
- Portfolio detail: https://haikalhilmi.my.id/projects/twitter-bot

Client testimonials

"Haikal supported me for nearly two years on multiple digital products. He quickly understood the business goals, proposed practical solutions, and worked independently across software development, integrations, databases, and data processing. I would gladly work with him again."

- Robert Ligthart, Founder of OmniSocials - Long-term SaaS Client; Amsterdam, Netherlands (profile (https://www.linkedin.com/in/robert-ligthart/))

"As commissioner of Semesta Data Digital and Global Data Inspirasi, I've seen first-hand how Haikal designs data infrastructure, runs large-scale pipelines and scraping with high reliability, and helped shape products like Robota. He was also my teaching assistant for a year - consistent, a fast learner, and highly dependable. He combines genuine technical depth with a strong sense of ownership."

- Widyawan, S.T., M.Sc., Ph.D., Commissioner, Semesta Data Digital & Global Data Inspirasi; Yogyakarta, ID (profile (https://www.linkedin.com/in/widyawan/))

"As Director of PT Aksarakan Bhumi Indonesia, I deeply appreciate his work - he not only built the company's website and digital infrastructure, but also drafted our IT security SOPs and actively took part in negotiations with strategic clients, showing a remarkable combination of technical skill, leadership, and business sense."

- Alamsyah Pangestu, Director, PT Aksarakan Bhumi Indonesia; Yogyakarta, ID (profile (https://www.linkedin.com/in/alamsyah-pangestu/))

"I had the pleasure of working with Haikal on a scraping project, and he exceeded my expectations in every aspect. Professional, responsive, and highly skilled, delivering high-quality work on time. I highly recommend Haikal - he's a fantastic contractor."

- Jeesun Kim, Upwork client - Twitter/X Scraper; Aurora, Canada

"Excellent service, quick responsiveness, and great technical skills. Haikal developed an effective and high-performing URL scraper, offered a 3-month warranty, and I would highly recommend him to others."

- Sam White, Upwork client - URL Scraper - 5.0 review; United States

"Haikal is very smart, creative and good at what he does. He is a great communicator and very efficient."

- Claire Bartolozzi, Upwork client - Tweet Collector; South Yarra, Australia

"Haikal was very quick to respond and professional in his work. He even suggested some improvements to the deliverables, which I found useful. I would definitely work with him again."

- Cristian Buda, Upwork client - PDF to JSON Extraction; Cluj-Napoca, Romania

Source and freshness

- Canonical portfolio: https://haikalhilmi.my.id
- Portfolio scope: general
- Markdown export: https://haikalhilmi.my.id/portfolio.md
- PDF export: https://haikalhilmi.my.id/portfolio.pdf
- AI discovery file: https://haikalhilmi.my.id/llms.txt
- The export is generated from the same typed project and experience data used by the website.