Meeting 14: Green AI & Sustainability

AI-40X: Generative AI & Large Language Models

Hendri Karisma, M.T.
Dosen Teknik Informatika STMIK Tazkia
VP Engineering at jejakin.com, 2026

Semester 4 — 2025/2026

Outline

Biaya Tersembunyi di Balik AI

- Setiap kali kita bertanya ke ChatGPT, ada **listrik** yang dikonsumsi oleh data center.
- Listrik tersebut menghasilkan **emisi karbon** (CO_2), terutama jika berasal dari bahan bakar fosil.
- AI bukan hanya soal akurasi model – tapi juga soal **tanggung jawab lingkungan**.

Fakta Mengejutkan

Satu query ChatGPT mengonsumsi $\sim$10x listrik dibandingkan satu pencarian Google biasa.

Jejak Karbon Training LLM

Model	Energi (MWh)	CO_2 (ton)
BERT (110M params)	1.5	0.6
GPT-2 (1.5B params)	27	11
GPT-3 (175B params)	∼1,287	∼552
Llama 2 70B	∼539	∼219

- Training GPT-3 menghasilkan ∼552 ton CO_2 – setara dengan **5 mobil selama masa pakainya** (masing-masing ∼110 ton CO_2).
- Ini belum termasuk eksperimen gagal, hyperparameter tuning, dan iterasi yang tidak dipublikasikan.

Training vs Inference: Mana yang Lebih Boros?

Training

- Sangat intensif: minggu–bulan di ribuan GPU.
- Tapi hanya dilakukan **sekali** (atau beberapa kali).
- Contoh: Training GPT-4 $\approx$ 3–6 bulan.

Inference

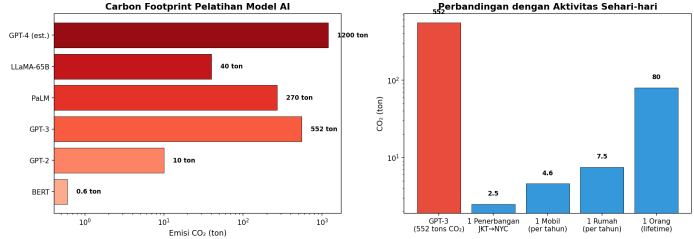
- Per-request kecil, tapi dilakukan **miliaran kali** per hari.
- Pada skala besar, inference mendominasi total energi.
- ChatGPT: $\sim$200 juta user, ratusan juta query/hari.

Estimasi

Pada skala deployment, $\sim$90% total energi AI berasal dari **inference**, bukan training.
Ini artinya: **optimasi inference sama pentingnya dengan optimasi training.**

Tren Konsumsi Energi Data Center

- Data center global mengonsumsi ∼1–2% listrik dunia (dan terus naik).
- Pertumbuhan AI diperkirakan meningkatkan konsumsi energi data center 2–3x lipat pada 2030.
- Perusahaan besar (Google, Microsoft, Meta) membangun data center baru khusus untuk AI.



Metrik Penting

- **kWh (Kilowatt-hour):** Satuan energi listrik yang dikonsumsi.
- **kg CO_2eq:** Kilogram karbon dioksida ekuivalen – mengukur dampak gas rumah kaca.
- **PUE (Power Usage Effectiveness):** Rasio total energi data center dibagi energi yang dipakai komputer.
  - PUE = 1.0 (sempurna, semua listrik untuk komputasi).
  - PUE = 1.2 (baik, 20% overhead untuk pendingin dll).
  - PUE = 2.0 (buruk, setengah listrik terbuang).
- **Carbon Intensity:** Emisi CO_2 per kWh listrik – bergantung pada sumber energi (batubara vs angin vs surya).

Lab: Mengukur Emisi Kode Anda dengan CodeCarbon

Menggunakan CodeCarbon

```
Mulai tracking tracker = EmissionsTracker(
project_name = "sentiment - analysis - tazkia", output_dir = "./emissions/", country_iso_code = "IDN" Indonesia)tracker.start()
Jalankan model classifier = pipeline("sentiment-analysis", model="indobenchmark/indobert-base-p1") results = classifier(["Kampus ini
sangat bagus!"] * 1000)
Stop tracking dan lihat hasilnya emissions = tracker.stop() print(f"Energi:
tracker._total_energy.kWh : .6fkWh")print(f" Emisi : emissions : .6f kgCO2eq")
```

- **MLCo2 Calculator** (`https://mlco2.github.io/impact/`): Estimasi emisi tanpa menjalankan kode.

Quantization: Mengecilkan Presisi Angka

- Setiap parameter model disimpan sebagai angka floating point.
- Mengurangi presisi = mengurangi ukuran model = lebih hemat energi dan memori.

Format	Bit/param	Ukuran 7B Model	Akurasi Relatif
FP32 (full)	32	28 GB	100% (baseline)
FP16 / BF16	16	14 GB	$\sim$99.5%
INT8	8	7 GB	$\sim$98%
INT4 (GPTQ/GGUF)	4	3.5 GB	$\sim$95%

Dampak

Model 7B dari 28 GB $\rightarrow$ 3.5 GB (INT4): bisa dijalankan di laptop biasa!
Pengurangan energi inference hingga 4x lipat.

Pruning & Knowledge Distillation

Pruning (Pemangkasan)

- Menghapus weight/neuron yang nilainya mendekati nol (tidak berkontribusi signifikan).
- **Unstructured Pruning:** Menghapus weight individual (sparse matrix).
- **Structured Pruning:** Menghapus seluruh neuron/head/layer.
- Bisa mengurangi parameter 50–90% dengan penurunan akurasi minimal.

Knowledge Distillation

- Melatih model kecil (*student*) untuk meniru perilaku model besar (*teacher*).
- Student tidak belajar dari data asli, tapi dari **distribusi probabilitas output** teacher.
- Contoh: DistilBERT (66M) meniru BERT (110M) dengan 97% akurasi dan 60% lebih cepat.
- Contoh LLM: Mistral 7B $\rightarrow$ student model 1.5B.

Arsitektur Efisien: MoE & Flash Attention

Mixture of Experts (MoE)

- Model punya banyak "expert" (sub-network), tapi hanya **sebagian kecil** yang aktif per token.
- Contoh: Mixtral 8x7B memiliki 46.7B total parameter, tapi hanya 12.9B yang aktif per inference.
- Hasil: kualitas setara model 46B, tapi kecepatan dan biaya mendekati model 13B.

Flash Attention

- Attention standar: $O(n^2)$ memori untuk sequence length n.
- Flash Attention (Dao, 2022): menghitung attention dengan **tiling** – memecah matriks besar menjadi blok kecil yang muat di SRAM GPU.
- Hasilnya: 2–4x lebih cepat, memori berkurang drastis.
- Memungkinkan context window lebih panjang (128K+ token).

Right-Sizing: Pilih Model Sesuai Kebutuhan

- Tidak semua tugas butuh GPT-4 atau model 70B.
- Prinsip: **gunakan model terkecil yang bisa menyelesaikan tugas dengan baik**.

Tugas	Model yang Cukup
Klasifikasi sentimen	BERT / DistilBERT (110–66M)
Ekstraksi entitas (NER)	Model fine-tuned kecil (<1B)
Chatbot FAQ sederhana	Llama 3 8B / Mistral 7B
Summarization	Llama 3 8B (quantized INT4)
Complex reasoning, coding	GPT-4o / Claude / Llama 3 70B
Creative writing, roleplay	Model besar (70B+)

Aturan Praktis

Mulai dari model kecil, evaluasi performanya. Naik ke model besar **hanya jika diperlukan**.

Perbandingan Biaya: API vs Self-Hosted

Opsi	Biaya/1M token	Latency	Privasi
GPT-4o (API)	$5–15	Rendah	Data ke OpenAI
GPT-4o-mini (API)	$0.15–0.6	Rendah	Data ke OpenAI
Llama 3 70B (cloud GPU)	$1–3	Sedang	Terkontrol
Llama 3 8B (self-hosted)	$0.1–0.5	Sedang	Penuh
Llama 3 8B INT4 (laptop)	Listrik saja	Tinggi	Penuh

- Untuk **startup/UMKM Indonesia**: model kecil self-hosted seringkali paling ekonomis.
- Untuk **prototyping**: API (GPT-4o-mini) paling cepat memulai.

Edge Deployment: AI di Perangkat Kecil

- **Edge AI** = menjalankan model langsung di perangkat end-user (smartphone, Raspberry Pi, IoT).
- Keuntungan:
  - Tidak butuh koneksi internet.
  - Privasi data terjaga (data tidak keluar perangkat).
  - Latency sangat rendah (real-time).
- Teknologi pendukung:
  - **ONNX Runtime:** Format model universal yang optimal.
  - **TensorFlow Lite / Core ML:** Untuk Android dan iOS.
  - **llama.cpp / Ollama:** Menjalankan LLM di CPU.
  - **Quantization INT4:** Wajib untuk perangkat dengan RAM terbatas.

Prinsip Pengembangan AI Berkelanjutan

1. **Reuse > Retrain > Train from Scratch**
   - Gunakan model pre-trained dari Hugging Face Hub.
   - Fine-tune dengan LoRA/QLoRA (hemat 90%+ energi vs full training).
   - Training dari nol hanya jika benar-benar diperlukan.
2. **Transfer Learning sebagai Strategi Sustainability**
   - Satu model besar (GPT, Llama) dilatih sekali oleh perusahaan besar.
   - Jutaan developer melakukan fine-tuning ringan untuk kebutuhan spesifik.
   - Jauh lebih efisien daripada setiap orang melatih dari nol.
3. **Model Sharing & Open Source**
   - Hugging Face Hub: 500K+ model tersedia gratis.
   - Berbagi model = mengurangi duplikasi pelatihan = mengurangi emisi global.

Green Computing & Carbon-Aware Scheduling

Green Data Center

- Google: 100% renewable energy sejak 2017 (matching).
- Microsoft: carbon negative target 2030.
- Lokasi strategis: Islandia, Norwegia (geothermal + hydro + pendinginan alami).
- Inovasi: liquid cooling, waste heat recovery.

Carbon-Aware Scheduling

- Jadwalkan komputasi berat saat **grid listrik paling bersih** (siang hari = banyak solar).
- Tools: `electricity-map.org` – peta real-time carbon intensity.
- Contoh: training di malam hari (Eropa) vs siang hari (daerah kaya solar) bisa beda emisi 2–5x.

Hirarki Sustainability AI

Reuse (pakai model yang sudah ada) → **Fine-tune** (adaptasi ringan) → **Retrain** (training ulang penuh) → **Train from scratch** (opsi terakhir).

Bauran Energi Indonesia

- Indonesia masih sangat bergantung pada **bahan bakar fosil**:
  - Batubara: $\sim$40%
  - Gas alam: $\sim$15%
  - Minyak bumi: $\sim$30%
  - Energi terbarukan: $\sim$15% (hydro, geothermal, solar)
- **Carbon intensity** listrik Indonesia: $\sim$0.7–0.8 kg CO_2/kWh (termasuk tinggi secara global).
- Artinya: **setiap kWh komputasi AI di Indonesia menghasilkan emisi lebih tinggi** dibandingkan di negara dengan energi bersih (Prancis: $\sim$0.06 kg CO_2/kWh).

Implikasi

Efisiensi model bahkan lebih penting di Indonesia karena carbon intensity yang tinggi.

Peluang: Lompatan ke Efficient AI

- Indonesia tidak perlu mengikuti jalur "model makin besar" seperti Silicon Valley.
- **Leapfrog Opportunity:** Langsung adopsi teknik efisien.
  - Small models (1–8B) yang di-quantize.
  - Edge deployment untuk daerah tanpa internet stabil.
  - Fine-tuning LoRA pada data lokal (Bahasa Indonesia, dialek daerah).
- **Digital Divide:**
  - 40%+ penduduk Indonesia belum punya akses internet stabil.
  - AI berbasis cloud tidak menjangkau mereka.
  - Solusi: AI on-device (smartphone murah + model kecil).

Studi Kasus: AI untuk Indonesia

Pertanian

- Deteksi hama tanaman via kamera smartphone.
- Model: MobileNet (klasifikasi gambar, 3–5 MB).
- Berjalan offline di sawah tanpa internet.
- Dampak: petani kecil bisa diagnosis sendiri tanpa ahli.

Kesehatan

- Screening awal penyakit mata (retinopati diabetik) di puskesmas.
- Model kecil + kamera fundus portable.
- Mengurangi kebutuhan dokter spesialis di daerah terpencil.

Pendidikan

- Tutor AI offline untuk daerah 3T (tertinggal, terdepan, terluar).
- Model bahasa kecil yang bisa menjawab pertanyaan pelajaran.
- Berjalan di tablet tanpa cloud.
- Personalized learning untuk setiap siswa.

Pesan Utama

AI yang berdampak besar tidak harus model yang besar. Model kecil yang tepat sasaran bisa mengubah kehidupan jutaan orang.

Kesimpulan

1. AI memiliki **jejak karbon yang signifikan**: training GPT-3 $\approx$ 552 ton CO_2.
2. **Inference mendominasi** total emisi pada skala deployment ($\sim$90%).
3. Kita bisa mengukur emisi kode kita dengan tools seperti **CodeCarbon**.
4. Teknik efisiensi (**quantization, pruning, distillation, MoE, Flash Attention**) bisa mengurangi energi hingga 4–10x.
5. Prinsip utama: **Reuse** $>$ **Retrain** $>$ **Train from scratch**.
6. Konteks Indonesia: carbon intensity tinggi $\rightarrow$ efisiensi model **lebih penting lagi**.
7. **Small models + edge deployment** = kunci AI yang inklusif dan berkelanjutan untuk Indonesia.

Pesan Penutup

"The greenest computation is the one you don't have to do."

— Prinsip Green AI

- Sebagai calon AI Engineer, kita punya **tanggung jawab** untuk membangun AI yang tidak hanya cerdas, tapi juga **bertanggung jawab terhadap lingkungan**.
- Mulai dari hal kecil: pilih model yang tepat, ukur emisi, dan bagikan model Anda.

Referensi

- Strubell, E., Ganesh, A., & McCallum, A. (2019). "Energy and Policy Considerations for Deep Learning in NLP." *ACL 2019*.
- Patterson, D., et al. (2021). "Carbon Emissions and Large Neural Network Training." *Google Research*.
- Schwartz, R., et al. (2020). "Green AI." *Communications of the ACM*.
- Dao, T., et al. (2022). "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness." *NeurIPS 2022*.
- CodeCarbon: `https://codecarbon.io`
- ML CO2 Impact Calculator: `https://mlco2.github.io/impact/`