Dokumen KP Blockchain – Pallet Audit untuk Audit Train Model

Bagian dari Project KP : Skor Kelayakan Penerima Subsidi Tanpa Tukar Data: Federated Learning dan Audit On-Chain

Penyusun : Imanuel P. A Faot

Ringkasan Project

Proyek ini ingin membuat audit on chain kontribusi perubahan model federated learning untuk proses kolaborasi antar akun tanpa pertukaran data atau sesuai dengan kasus yaitu institusi. Setiap hasil pelatihan/agregasi atau perubahan (model) direpresentasikan sebagai hash kriptografis dan dicatat on-chain melalui pallet-audit di jaringan Substrate, dengan tujuan riwayat kontribusi dapat dibuktikan, ditelusuri, dan diaudit dengan baik tanpa duplikat serta tidak bisa dimanipulasi dan transparan.

Masalah

- Setiap node atau tempat atau instansi melakukan train model federated learning dengan data lokal mereka masing-masing
- Aturan UU PDP sehingga tidak boleh ada data yang diserahkan ke server
- Tidak ada pencatatan siapa yang melakukan perubahan dan apa yang berubah atau diubah
- Pengarsipan pencatatan perubahan model yang rawan dimanipulasi

Solusi

- Menyimpan hash kontribusi perubahan model serta metadata setiap kali ada aksi perubahan model dan secara immutable atau tidak bisa diubah
- Menghasilkan event setiap ada pencatatan kontribusi model serta menyediakan query indeks untuk mengakses pencatatan tersebut

Batasan prototype dokumen ini

- Lokal dengan solochain template
- Menjalankan pallet audit sendiri pada solochain
- Antarmuka menggunakan [Polkadot/Substrate Portal]
- Penyimpanan dengan IPFS opsional ketika perubahan weight model terlalu besar

Requirement

- Fungsional
    - Peran akses

Admin (origin = AuthorityOrigin, default Root/Sudo)	force_authorize(account, institution) ● memasang/memperbarui otorisasi AccountId → InstName.
	force_unauthorize(account) ● mencabut otorisasi.
Operator (akun biasa yang diotorisasi)	submit(model_hash, ipfs_cid?, note?) ● mencatat satu audit record. Institusi diambil otomatis dari whitelist.
Viewer	Query state: ● by_hash(model_hash) -> Option<Id> ● records(Id) -> AuditRecord ● by_account(AccountId) -> [Id], ● by_institution(InstName) -> [Id]

    - Validasi & Proteksi
        - Anti-duplikasi hash: satu model_hash hanya boleh 1 Id (error AlreadyExists).
        - Hanya akun yang di-whitelist yang boleh submit (error NotAuthorized). Note : institution tidak berasal dari input user (bukan argumen extrinsic); diset otomatis dari AuthorizedInstitution.

        - institution tidak boleh kosong saat otorisasi (error EmptyInstitution).
        - Batas kapasitas indeks:
            - by_account max MaxPerAccount (error TooManyForAccount)
            - by_institution max MaxPerInstitution (error TooManyForInstitution).
    - Event

AuthorizedSet { account, institution }	AuthorizedRemoved { account }	Submitted { id, who, model_hash, institution, at_block, ipfs_cid }

    - Acceptance Criteria
        - force_authorize menambah entry; event AuthorizedSet terbit; authorized_institution(account) = institusi.
        - submit dari akun terotorisasi menghasilkan event Submitted dan records(id) konsisten.
        - submit duplikat model_hash gagal dengan AlreadyExists.
        - submit dari akun tidak terotorisasi gagal NotAuthorized.
        - Query by_hash : Some(id), records(id), by_account, by_institution berfungsi.

- Non-fungsional

Keamanan & Auditability	• Transaksi ditandatangani (sr25519/ed25519) → who terverifikasi. • Jejak immutable (append-only); setiap aksi penting memicu **event**. • Institusi dipatok dari whitelist admin (menghindari manipulasi input).
Kinerja & Skala	• Parameter skala dikendalikan lewat konstanta: MaxPerAccount, MaxPerInstitution, Max*Len.
Kompatibilitas	• Kompatibel dengan Polkadot.js Apps & Subxt. • ModelHash = H256 (32 byte); ipfs_cid & note bertipe Bytes bounded.

Privasi	• On-chain hanya menyimpan hash + metadata kecil; artefak (model) off-chain. • Memakai IPFS, dapat menyimpan CID artefak/metadata; enkripsi artefak direkomendasikan bila sensitif.

- Operasional
  - Konfigurasi Runtime

```
impl pallet_audit::Config for Runtime:

type ModelHash = sp_core::H256
type AuthorityOrigin = frame_system::EnsureRoot<AccountId> (atau
kebijakan lain: EnsureOneOf<Root, Admins/Membership>).
```

  - Konstanta:

```
MaxInstNameLen (disarankan ≥ 64)
MaxNoteLen (mis. 256)
MaxCidLen (mis. 64)
MaxPerAccount (mis. 500)
MaxPerInstitution (mis. 10_000)
type WeightInfo = () untuk demo (produksi: hasil benchmarking).
```

- Data
  - Struktur Data

```
AuditRecord {
  id: u64,                        // auto-increment
  who: AccountId,                 // pengirim (penanda tangan)
  at_block: BlockNumber,          // cap waktu berbasis blok
  model_hash: H256,               // jejak artefak (unik)
  institution: BoundedVec<u8>,    // snapshot institusi saat submit
  ipfs_cid: Option<BoundedVec<u8>>, // CID v1 base32 (opsional)
  note: Option<BoundedVec<u8>>,      // catatan singkat (opsional)
}
```

  - Storage

```
NextId: Id
```

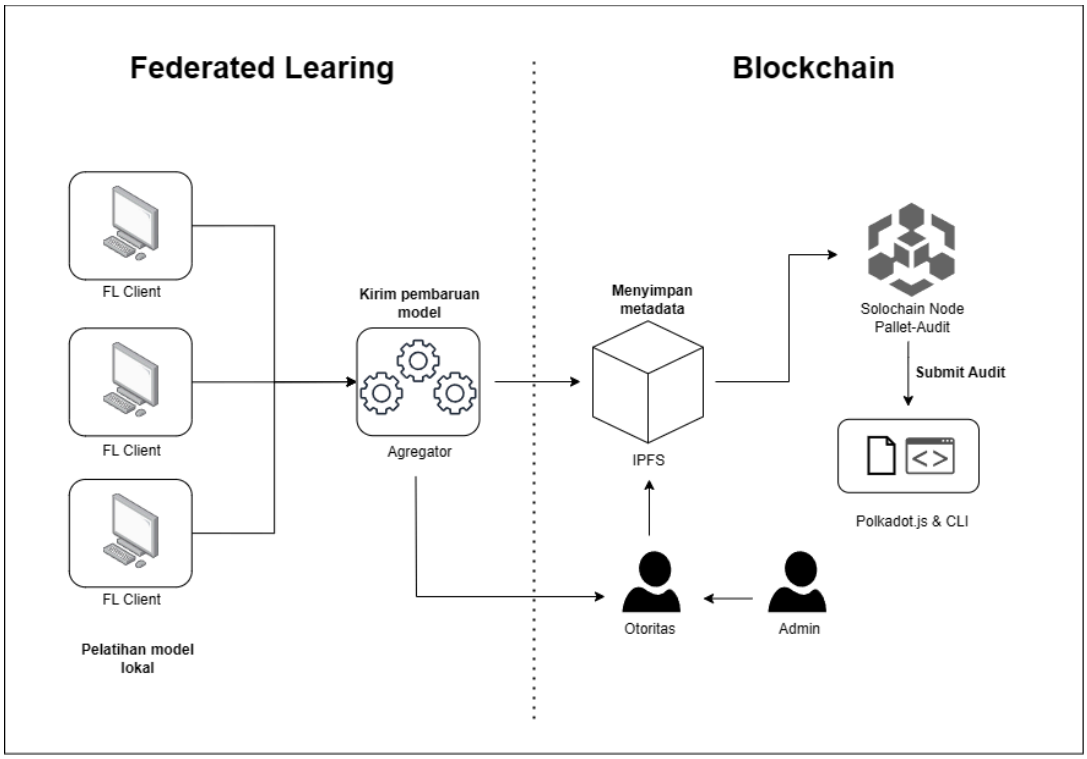
```
Records: Id -> AuditRecord
ByHash: ModelHash -> Id (unik)
ByAccount: AccountId -> BoundedVec<Id, MaxPerAccount>
ByInstitution: InstName -> BoundedVec<Id, MaxPerInstitution>
AuthorizedInstitution: AccountId -> InstName
```

- Batas
  - InstNameLen ≤ MaxInstNameLen (disarankan 64).
  - CidLen ≤ MaxCidLen (disarankan 64).
  - NoteLen ≤ MaxNoteLen (disarankan 256).

- Aturan Integritas
  - Satu model_hash → satu Id (unik).
  - institution disalin dari whitelist (snapshot) saat submit → riwayat afiliasi tetap valid meski akun berpindah institusi di masa depan.

- Data Off-chain
  - Artefak model tidak disimpan on-chain.
  - Rekomendasi: hash deterministik (SHA-256), arsip deterministik untuk folder (tar + sort + fixed mtime), pinning IPFS.

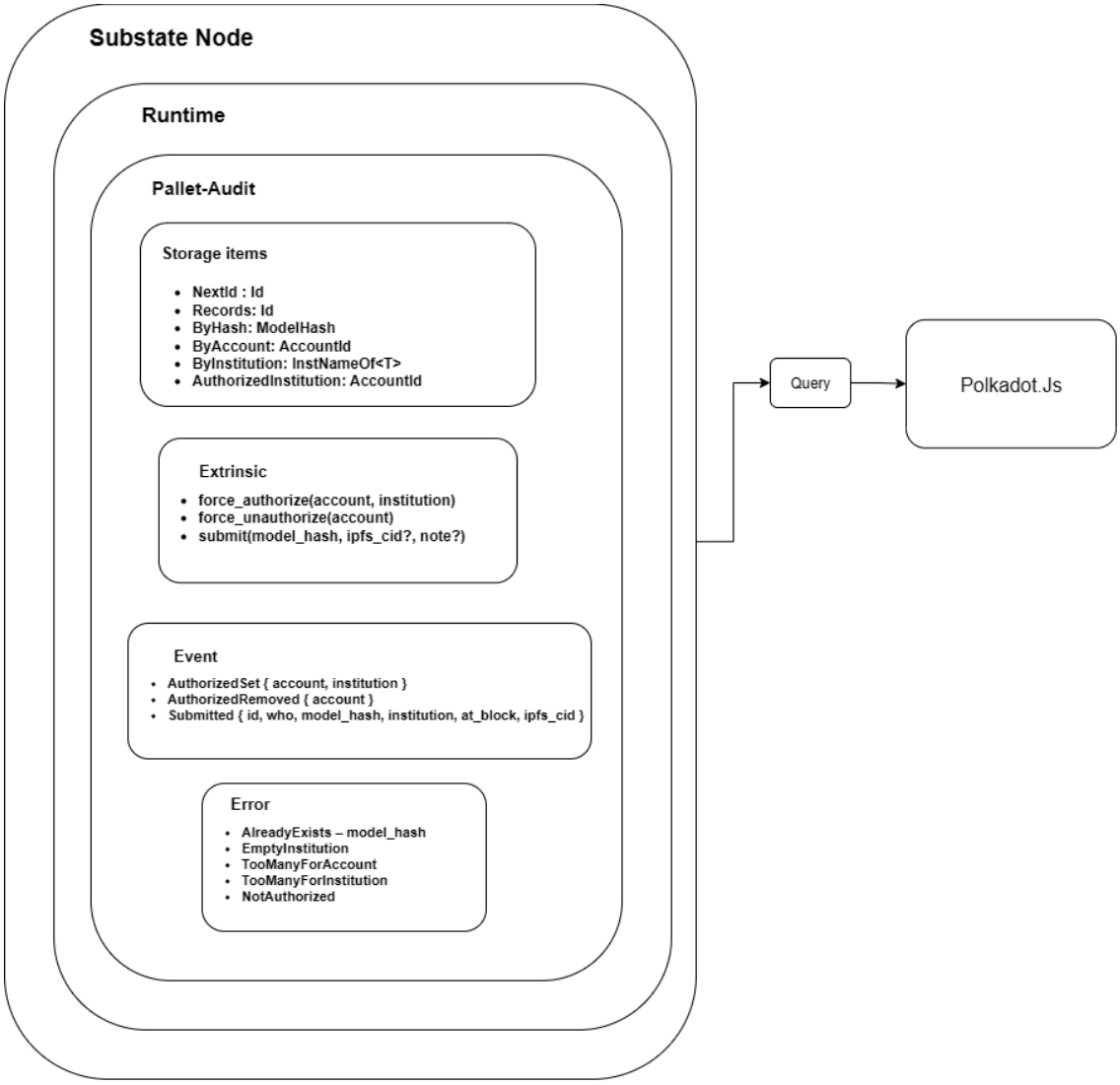
Diagram Sistem



Peran

Role	Deskripsi
FL Client (Instansi)	Melatih model lokal & mengirim pembaruan ke agregator.
Agregator/Orkestrator	Menggabungkan pembaruan.
Admin (Sudo/Root)	Mengotorisasi akun institusi yang boleh mencatat.
Otoritas	Unggah ke IPFS, dan mengirim extrinsic audit.submit.
Blockchain (Solochain Node + pallet-audit)	Menyimpan jejak; menyediakan event & query.
Auditor/Viewer (Polkadot.js/CLI/Dashboard)	Verifikasi dan validasi riwayat.

Arsitektur Blockchain



Alur kerja

1. Persiapan (Pra-kondisi)
    1. Admin menetapkan otorisasi akun:
        - sudo.sudo(audit.forceAuthorize(account=AKUN_INSTANSI, institution="NAMA-INSTANSI")).

    2. Agregator berjalan (mis. Flower/NVFLARE/OpenFL).
    3. (Opsional) IPFS node/pinning siap digunakan.
    4. Kunci akun institusi aman (sr25519/ed25519).

2. Siklus Federated Learning
    1. FL Client di masing-masing instansi melatih model lokal (data tidak keluar lokasi).
    2. FL Client mengirim pembaruan model ke Aggregator.
    3. Agregator menjalankan agregasi dan menghasilkan artefak model utama.
    4. Agregator menyimpan artefak ke storage lokal (atau object storage internal).

3. Penerbitan Jejak (Hashing → IPFS → Submit)
    1. Hasher/Publisher membaca artefak model final dan menghitung

        SHA-256 → MODEL_HASH (H256).

    2. (Opsional) Hasher/Publisher mengunggah artefak/metadata ke IPFS → memperoleh CID.
    3. Hasher/Publisher (menggunakan akun institusi yang telah diotorisasi) mengirim extrinsic:
        - audit.submit(modelHash=MODEL_HASH, ipfsCid=Some(CID)? , note=Some("round-7")?).
        - Catatan: institution tidak diinput; pallet mengambilnya otomatis dari whitelist.

4. Pencatatan On-chain (pallet-audit)
    1. Pallet memverifikasi:
        - Akun terdaftar di AuthorizedInstitution : jika tidak, gagal NotAuthorized.
        - MODEL_HASH belum ada → jika duplikat, gagal AlreadyExists.
    2. Jika valid, pallet:
        - Membuat record: {id, who, at_block, model_hash, institution, ipfs_cid?, note?}.
        - Memperbarui indeks: ByHash, ByAccount, ByInstitution.

- Memancarkan event Audit.Submitted.

5. Verifikasi & Audit
    1. Auditor/Viewer membuka Polkadot.js/CLI/Dashboard:

byHash(MODEL_HASH): Some(id).
records(id) : menampilkan who, institution, at_block, ipfs_cid, note.
byAccount(AKUN_INSTANSI) / byInstitution("NAMA-INSTANSI") → daftar id.

    2. Verifikasi off-chain:
        - Unduh artefak dari IPFS (jika CID tersedia), hitung SHA-256 lokal, bandingkan dengan MODEL_HASH di chain.

6. Penanganan Pengecualian (Exception Flow)

NotAuthorized	Admin jalankan forceAuthorize, ulangi submit.
NotAuthorized	Akun belum di-whitelist
AlreadyExists	Periksa apakah artefak memang sama; jika memang round baru, pastikan file/arsip dibangkitkan deterministik (hindari metadata berbeda).
Panjang nama institusi	Gunakan nama/kode singkat; atau naikkan parameter di runtime.
1010 Bad signature	Pastikan men-sign dengan akun Substrate (sr25519) dan hapus ekstensi unik (mis. CheckMetadataHash) pada lingkungan dev.

7. Pasca-kondisi & Pelaporan
    - Record bertambah satu, dengan snapshot institusi saat submit (meski afiliasi akun berubah di masa depan).
    - Log event tersedia untuk monitoring.
    - Dashboard (bila ada) otomatis menampilkan rekap kontribusi per institusi/akun/round.

8. Artefak Data

On-chain	`AuditRecord { id, who, at_block, model_hash(H256), institution(Bytes), ipfs_cid?(Bytes), note?(Bytes) }, indeks ByHash/ByAccount/ByInstitution.`
Off-chain	file model/arsip (opsional di-IPFS), metadata round (lokal).