Sept 25

**Pemetaannya :**

* **A. Policy Control**: MetadataGovernanceGoal, **MetadataStrategy**, MetadataPolicy, MetadataRule, MetadataStandard, MetadataQuality, ComplianceSource, MetadataAccessControl
* **B. Organizational Accountability**: MetadataOrganization, MetadataStewardship, MetadataOwner, OwnerRole
* **C. Assurance & Risk Control**: MetadataAudit, MetadataRisk, KPIIndicator
* **D. Governance Execution**: MetadataChangeRequest, MetadataIssueLog, LifecyclePhase

 **Diagram 4-klaster (draw.io) – 26 relasi + kardinalitas:**  
**Metamodel\_26\_labeled\_4clusters.drawio**

 **Katalog relasi (Excel) – 26 relasi, satu kolom relasi bilingual:**  
**Katalog\_Relasi\_26\_Bilingual.xlsx**

| **#** | **Relation (EN (IDN))** | **Type** | **Source→Target** | **Cardinality** | **OCL (brief)** | **Ontological Justification** | **Example Instantiation** | **References** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Policy-Defines-Standard (Kebijakan-Mendefinisikan-Standar) | Association | MetadataPolicy → MetadataStandard | 1..\* ↔ 1..\* | context MetadataPolicy inv: self.standards->size()>=1 | Policies normatively yield technical standards; not inheritance. | National Metadata Policy → ISO/IEC 11179 | DAMA-DMBOK2; ISO/IEC 11179 |
| 2 | Policy-Defines-Rule (Kebijakan-Mendefinisikan-Aturan) | Association | MetadataPolicy → MetadataRule | 1..\* ↔ 1..\* | context MetadataPolicy inv: self.rules->size()>=1 | Operational rules are concrete derivations of policies. | Policy “Naming” → Rule “≤50 chars” | DAMA-DMBOK2 |
| 3 | Policy-Ensures-Quality (Kebijakan-Memastikan-Kualitas) | Association | MetadataPolicy → MetadataQuality | 1..\* ↔ 0..\* | context MetadataPolicy inv: self.qualityTargets->notEmpty() | Policies set target values on quality dimensions. | Quality Policy → completeness ≥95% | DAMA-DMBOK2 ch.13 |
| 4 | Goal-Guides-Policy (Tujuan-Mengarahkan-Kebijakan) | Association | MetadataGovernanceGoal → MetadataPolicy | 1 ↔ 1..\* | context MetadataPolicy inv: self.goal->notEmpty() | Goals orient the formation of policies (teleology→normativity). | Interoperability goal → standardization policy | ISO/IEC 38500 |
| 5 | Compliance-Constrains-Policy (SumberKepatuhan-Membatasi-Kebijakan) | Association | ComplianceSource → MetadataPolicy | 1..\* ↔ 1..\* | context MetadataPolicy inv: self.complianceSources->size()>=1 | External regulations bound and inform policies. | GDPR → Retention Policy | GDPR; ISO/IEC 27001 |
| 6 | Compliance-Constrains-Rule (SumberKepatuhan-Membatasi-Aturan) | Association | ComplianceSource → MetadataRule | 1..\* ↔ 1..\* | context MetadataRule inv: self.complianceSources->size()>=1 | Specific rules often stem from regulations. | GDPR → storage/retention rule | GDPR |
| 7 | Rule-Requires-AccessControl (Aturan-Memerlukan-KontrolAkses) | Association | MetadataRule → MetadataAccessControl | 1..\* ↔ 1..\* | context MetadataRule inv: self.accessControls->size()>=1 | Rule enforcement needs access controls (role/data-class). | Attribute-level rule → ACL/ABAC | ISO/IEC 27001 |
| 8 | Rule-VerifiedBy-Audit (Aturan-DiverifikasiOleh-Audit) | Association | MetadataRule → MetadataAudit | 1..\* ↔ 1..\* | context MetadataRule inv: self.audits->size()>=1 | Audit checks rule compliance. | Audit 2024 checks naming rule | COBIT 2019; ISO/IEC 27001 |
| 9 | Organization-Assigns-Stewardship (Organisasi-Menetapkan-Stewardship) | Association | MetadataOrganization → MetadataStewardship | 1 ↔ 1..\* | context MetadataOrganization inv: self.stewardships->size()>=1 | Stewardship is an organizational mandate/appointment. | BPS appoints Metadata Catalog Stewardship | DAMA-DMBOK2 |
| 10 | Stewardship-Has-Owner (Stewardship-Memiliki-Pemilik) | Association | MetadataStewardship → MetadataOwner | 1..\* ↔ 1..1 | context MetadataStewardship inv: self.owner->size()=1 | Each stewardship has a single accountable owner. | Stewardship A → Owner “Director of Statistics” | Internal policy |
| 11 | Owner-Has-Role (Pemilik-Memiliki-Peran) | Association | MetadataOwner → OwnerRole | 1 ↔ 1..\* | context MetadataOwner inv: self.roles->notEmpty() | An owner may play one or more formal roles. | Owner X → Data Custodian | ISO/IEC 38500 |
| 12 | Owner-Oversees-Quality (Pemilik-Mengawasi-Kualitas) | Association | MetadataOwner → MetadataQuality | 1 ↔ 1..\* | context MetadataOwner inv: self.qualityTargets->notEmpty() | Quality accountability is assigned to owners. | Owner sets completeness targets | DAMA-DMBOK2 |
| 13 | Owner-Authorizes-Change (Pemilik-Mengesahkan-Perubahan) | Association | MetadataOwner → MetadataChangeRequest | 1 ↔ 0..\* | context MetadataChangeRequest inv: self.authorizedBy->size()<=1 | Definition changes must be authorized by owner. | Owner approves definition change | COBIT 2019 |
| 14 | Owner-Resolves-Issue (Pemilik-Menyelesaikan-Isu) | Association | MetadataOwner → MetadataIssueLog | 1 ↔ 0..\* | context MetadataIssueLog inv: self.resolvedBy->size()<=1 | Metadata issues are resolved by the relevant owner. | Owner resolves validation issue | DAMA-DMBOK2 |
| 15 | Audit-Validates-Policy (Audit-Memvalidasi-Kebijakan) | Association | MetadataAudit → MetadataPolicy | 1..\* ↔ 1..\* | context MetadataAudit inv: self.policies->notEmpty() | Audits validate policy conformance. | Policy compliance audit | ISO/IEC 27001; ISO 19011 |
| 16 | Audit-Reports-KPI (Audit-Melaporkan-KPI) | Association | MetadataAudit → KPIIndicator | 1..\* ↔ 0..\* | context MetadataAudit inv: self.kpis->notEmpty() | Audits produce indicators for governance reporting. | 2024 audit → 90% compliance KPI | COBIT 2019 |
| 17 | Risk-Impacts-Quality (Risiko-BerdampakPada-Kualitas) | Association | MetadataRisk → MetadataQuality | 1..\* ↔ 1..\* | context MetadataRisk inv: self.affectedQuality->notEmpty() | Risks affect quality dimensions. | Integrity risk → consistency | ISO/IEC 27001 |
| 18 | Risk-MonitoredBy-Audit (Risiko-DimonitorOleh-Audit) | Association | MetadataRisk → MetadataAudit | 1..\* ↔ 1..\* | context MetadataRisk inv: self.monitoringAudits->notEmpty() | Risk controls are monitored by audit. | Risk mitigation audit | DAMA-DMBOK2; ISO/IEC 27001 |
| 19 | ChangeRequest-Relates-Issue (PermintaanPerubahan-Terkait-Isu) | Association | MetadataChangeRequest → MetadataIssueLog | 1..\* ↔ 0..\* | context MetadataChangeRequest inv: self.issues->notEmpty() | Change requests are typically triggered by recorded issues. | Validation issue → CR #123 | DAMA-DMBOK2 |
| 20 | Lifecycle-Impacts-Change (SiklusHidup-BerdampakPada-Perubahan) | Association | LifecyclePhase → MetadataChangeRequest | 1..\* ↔ 0..\* | context LifecyclePhase inv: self.changeRequests->notEmpty() | Lifecycle phases drive change needs. | Decommission → deletion CR | ISO/IEC 11179 |
| 21 | Lifecycle-TrackedBy-Audit (SiklusHidup-DilacakOleh-Audit) | Association | LifecyclePhase → MetadataAudit | 1..\* ↔ 0..\* | context LifecyclePhase inv: self.audits->notEmpty() | Each lifecycle phase has audit checkpoints. | Publication-phase audit | DAMA-DMBOK2 |
| 22 | KPI-Measures-Goal (KPI-Mengukur-Tujuan) | Association | KPIIndicator → MetadataGovernanceGoal | 1..\* ↔ 1..\* | context KPIIndicator inv: self.goal->notEmpty() | KPIs measure attainment of governance goals. | Compliance KPI → Quality goal | COBIT 2019 |
| 23 | Organization-Performs-Audit (Organisasi-Melaksanakan-Audit) | Association | MetadataOrganization → MetadataAudit | 1 ↔ 0..\* | context MetadataAudit inv: self.performedBy->size()=1 | Audits are executed/led by the organization (actor). | Compliance Unit → 2024 Audit | ISO/IEC 27001; ISO 19011; COBIT 2019 |
| 24 | Organization-Governs-Lifecycle (Organisasi-Mengatur-SiklusHidup) | Association | MetadataOrganization → LifecyclePhase | 1 ↔ 0..\* | context LifecyclePhase inv: self.governedBy->size()=1 | Organization governs execution of lifecycle phases. | Data Division → publication & decommission | ISO/IEC 11179; DAMA-DMBOK2; ISO/IEC 38500 |
| 25 | Goal-Guides-Strategy (Tujuan-Mengarahkan-Strategi) | Association | MetadataGovernanceGoal → MetadataStrategy | 1..\* ↔ 1..\* | context MetadataStrategy inv: self.goals->size()>=1 | Governance goals orient metadata/data strategy (goals cascade). | Goal 'Interoperability' → Strategy 'Enterprise Data Standardization' | COBIT 2019 (Goals Cascade); ISO/IEC 38500 |
| 26 | Strategy-Guides-Policy (Strategi-Mengarahkan-Kebijakan) | Association | MetadataStrategy → MetadataPolicy | 1..\* ↔ 1..\* | context MetadataPolicy inv: self.strategy->notEmpty() | Policies should be aligned with and guided by strategy (direct function). | Strategy 'Data Quality First' → Policy 'Completeness ≥95%' | ISO/IEC 38500 (Direct); COBIT 2019 APO01/APO02; DAMA-DMBOK2 |

**1) Policy–Defines–Standard**

Relasi ini menegaskan bahwa setiap kebijakan metadata harus diwujudkan dalam bentuk standar teknis. Secara ontologis, **Policy** berada pada tingkat prinsip/aturan normatif, sedangkan **Standard** mengoperasionalkan prinsip tersebut (aturan implementatif yang dapat diaudit).  
**Kardinalitas (disarankan).** Policy ↔ Standard = **1..**\* ↔ **1..**\* (setiap kebijakan menghasilkan ≥1 standar; setiap standar merujuk ≥1 kebijakan).  
**OCL (ringkas).** context MetadataPolicy inv: self.standards->size()>=1  
**Contoh.** Kebijakan Metadata Nasional menurunkan Standar ISO/IEC 11179 untuk penamaan elemen data.  
**Rujukan.** DAMA-DMBOK2; ISO/IEC 11179.

**2) Policy–Defines–Rule**

Relasi ini menyatakan bahwa aturan operasional (**Rule**) merupakan turunan konkrit dari kebijakan. Secara ontologis, **Rule** adalah constraint/prosedur teruji yang memampukan penilaian kepatuhan terhadap **Policy**.  
**Kardinalitas.** Policy ↔ Rule = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.rules->size()>=1  
**Contoh.** Kebijakan “Penamaan” → Rule “Panjang nama atribut ≤ 50 karakter”.  
**Rujukan.** DAMA-DMBOK2.

**3) Policy–Ensures–Quality**

Relasi ini menyatakan kebijakan menetapkan target pada dimensi kualitas metadata (**Quality**). Secara ontologis, **Policy** men-set nilai sasaran (accuracy, completeness, dsb.) yang dapat diuji.  
**Kardinalitas.** Policy ↔ Quality = **1..**\* ↔ **0..**\*.  
**OCL.** context MetadataPolicy inv: self.qualityTargets->notEmpty()  
**Contoh.** Kebijakan Kualitas → *completeness* ≥ 95%.  
**Rujukan.** DAMA-DMBOK2 (Bab Kualitas Data).

**4) Goal–Guides–Policy**

Relasi ini menyatakan **Governance Goal** mengarahkan pembentukan kebijakan. Ontologis: **Goal** (arah/teleologi) → **Policy** (normativitas).  
**Kardinalitas.** Goal ↔ Policy = **1** ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.goal->notEmpty()  
**Contoh.** *Goal* interoperabilitas → kebijakan standardisasi metadata.  
**Rujukan.** ISO/IEC 38500.

**5) Compliance–Constrains–Policy**

Relasi ini menyatakan sumber kepatuhan (regulasi/standar) **membatasi/mengarahkan** kebijakan. Ontologis: **ComplianceSource** adalah sumber normatif eksternal bagi **Policy**.  
**Kardinalitas.** Compliance ↔ Policy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.complianceSources->size()>=1  
**Contoh.** GDPR → Kebijakan Retensi.  
**Rujukan.** GDPR; ISO/IEC 27001.

**6) Compliance–Constrains–Rule**

Relasi ini menyatakan **Rule** spesifik sering diturunkan langsung dari regulasi. Ontologis: **ComplianceSource** memberi batas/persyaratan eksternal bagi **Rule**.  
**Kardinalitas.** Compliance ↔ Rule = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.complianceSources->size()>=1  
**Contoh.** GDPR → aturan “retention maksimum N tahun”.  
**Rujukan.** GDPR.

**7) Rule–Requires–AccessControl**

Relasi ini menyatakan penegakan **Rule** membutuhkan **AccessControl**. Ontologis: **AccessControl** adalah mekanisme kontrol pelaksana constraint **Rule**.  
**Kardinalitas.** Rule ↔ AccessControl = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.accessControls->size()>=1  
**Contoh.** Rule akses atribut sensitif → ACL/ABAC berbasis peran.  
**Rujukan.** ISO/IEC 27001.

**8) Rule–VerifiedBy–Audit**

Relasi ini menyatakan kepatuhan **Rule** diverifikasi melalui **Audit**. Ontologis: **Audit** adalah event evaluatif terhadap penerapan **Rule**.  
**Kardinalitas.** Rule ↔ Audit = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.audits->size()>=1  
**Contoh.** Audit 2024 memeriksa kepatuhan rule penamaan atribut.  
**Rujukan.** COBIT 2019; ISO/IEC 27001.

**9) Organization–Assigns–Stewardship**

Relasi ini menegaskan mandat **Stewardship** ditetapkan oleh **Organization** sebagai aktor institusional. Ontologis: agent (**Organization**) menunjuk peran/proses governance (**Stewardship**).  
**Kardinalitas.** Organization ↔ Stewardship = **1** ↔ **1..**\*.  
**OCL.** context MetadataOrganization inv: self.stewardships->size()>=1  
**Contoh.** BPS menetapkan *Stewardship* Katalog Metadata.  
**Rujukan.** DAMA-DMBOK2.

**10) Stewardship–Has–Owner**

Relasi ini menyatakan tiap **Stewardship** memiliki **Owner** akuntabel tunggal. Ontologis: **Owner** adalah *responsibility bearer* final.  
**Kardinalitas.** Stewardship ↔ Owner = **1..**\* ↔ **1..1**.  
**OCL.** context MetadataStewardship inv: self.owner->size()=1  
**Contoh.** Stewardship A → Owner “Direktur Statistik”.  
**Rujukan.** Kebijakan internal; praktik RACI.

**11) Owner–Has–Role**

Relasi ini menyatakan seorang **Owner** menjalankan satu atau lebih **OwnerRole** (Data Owner, Custodian, Steward). Ontologis: **Role** mengikat hak/otoritas/fungsi.  
**Kardinalitas.** Owner ↔ OwnerRole = **1** ↔ **1..**\*.  
**OCL.** context MetadataOwner inv: self.roles->notEmpty()  
**Contoh.** Owner X → *Role* Data Custodian.  
**Rujukan.** ISO/IEC 38500.

**12) Owner–Oversees–Quality**

Relasi ini menegaskan akuntabilitas kualitas metadata pada **Owner**. Ontologis: **Owner** sebagai pemikul akuntabilitas atas dimensi **Quality**.  
**Kardinalitas.** Owner ↔ Quality = **1** ↔ **1..**\*.  
**OCL.** context MetadataOwner inv: self.qualityTargets->notEmpty()  
**Contoh.** Owner menetapkan target *completeness*.  
**Rujukan.** DAMA-DMBOK2.

**13) Owner–Authorizes–Change**

Relasi ini menyatakan **ChangeRequest (CR)** harus disahkan **Owner**. Ontologis: **Owner** adalah *authority* pemberi legitimasi bagi **CR**.  
**Kardinalitas.** Owner ↔ ChangeRequest = **1** ↔ **0..**\*; dari CR ke Owner **0..1**.  
**OCL.** context MetadataChangeRequest inv: self.authorizedBy->size()<=1  
**Contoh.** Owner menyetujui CR pembaruan definisi elemen.  
**Rujukan.** COBIT 2019 (manajemen perubahan).

**14) Owner–Resolves–Issue**

Relasi ini menegaskan **IssueLog** metadata ditangani **Owner** terkait. Ontologis: **Owner** sebagai penanggung jawab *issue lifecycle*.  
**Kardinalitas.** Owner ↔ IssueLog = **1** ↔ **0..**\*; dari Issue ke Owner **0..1**.  
**OCL.** context MetadataIssueLog inv: self.resolvedBy->size()<=1  
**Contoh.** Owner menyelesaikan isu validasi definisi elemen.  
**Rujukan.** DAMA-DMBOK2.

**15) Audit–Validates–Policy**

Relasi ini menyatakan **Audit** menilai kepatuhan terhadap **Policy**. Ontologis: **Audit** adalah event verifikasi atas aturan normatif **Policy**.  
**Kardinalitas.** Audit ↔ Policy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataAudit inv: self.policies->notEmpty()  
**Contoh.** Audit Kepatuhan Kebijakan Metadata.  
**Rujukan.** ISO/IEC 27001; ISO 19011.

**16) Audit–Reports–KPI**

Relasi ini menyatakan **Audit** menghasilkan indikator kinerja (**KPIIndicator**) untuk pelaporan governance. Ontologis: **KPI** adalah artefak evaluatif dari **Audit**.  
**Kardinalitas.** Audit ↔ KPI = **1..**\* ↔ **0..**\*.  
**OCL.** context MetadataAudit inv: self.kpis->notEmpty()  
**Contoh.** Audit 2024 → KPI kepatuhan 90%.  
**Rujukan.** COBIT 2019.

**17) Risk–Impacts–Quality**

Relasi ini menyatakan **Risk** berdampak pada dimensi **Quality**. Ontologis: **Risk** sebagai potensi kejadian yang memengaruhi properti kualitas metadata.  
**Kardinalitas.** Risk ↔ Quality = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRisk inv: self.affectedQuality->notEmpty()  
**Contoh.** Risiko integritas → konsistensi metadata.  
**Rujukan.** ISO/IEC 27001.

**18) Risk–MonitoredBy–Audit**

Relasi ini menyatakan pengendalian **Risk** dimonitor melalui **Audit**. Ontologis: **Audit** menilai efektivitas kontrol terhadap **Risk**.  
**Kardinalitas.** Risk ↔ Audit = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRisk inv: self.monitoringAudits->notEmpty()  
**Contoh.** Audit mitigasi risiko metadata tahunan.  
**Rujukan.** DAMA-DMBOK2; ISO/IEC 27001.

**19) ChangeRequest–Relates–Issue**

Relasi ini menyatakan permintaan perubahan (**CR**) biasanya muncul dari **Issue** terdokumentasi. Ontologis: **Issue** sebagai sebab/justifikasi yang menautkan ke **CR**.  
**Kardinalitas.** ChangeRequest ↔ IssueLog = **1..**\* ↔ **0..**\*.  
**OCL.** context MetadataChangeRequest inv: self.issues->notEmpty()  
**Contoh.** Isu validasi → CR #123.  
**Rujukan.** DAMA-DMBOK2.

**20) Lifecycle–Impacts–Change**

Relasi ini menyatakan fase **Lifecycle** memicu kebutuhan **ChangeRequest**. Ontologis: **LifecyclePhase** adalah state/proses institusional yang menimbulkan **CR**.  
**Kardinalitas.** LifecyclePhase ↔ ChangeRequest = **1..**\* ↔ **0..**\*.  
**OCL.** context LifecyclePhase inv: self.changeRequests->notEmpty()  
**Contoh.** Fase dekomisioning → CR penghapusan metadata.  
**Rujukan.** ISO/IEC 11179.

**21) Lifecycle–TrackedBy–Audit**

Relasi ini menyatakan tiap fase **Lifecycle** memiliki checkpoint **Audit**. Ontologis: **Audit** mendokumentasi/memverifikasi transisi fase **Lifecycle**.  
**Kardinalitas.** LifecyclePhase ↔ Audit = **1..**\* ↔ **0..**\*.  
**OCL.** context LifecyclePhase inv: self.audits->notEmpty()  
**Contoh.** Audit pada fase publikasi metadata.  
**Rujukan.** DAMA-DMBOK2.

**22) KPI–Measures–Goal**

Relasi ini menyatakan **KPI** mengukur ketercapaian **Governance Goal**. Ontologis: **KPIIndicator** adalah predikat kuantitatif atas **Goal**.  
**Kardinalitas.** KPI ↔ Goal = **1..**\* ↔ **1..**\*.  
**OCL.** context KPIIndicator inv: self.goal->notEmpty()  
**Contoh.** KPI “Kepatuhan 90%” → Goal “Kualitas Metadata”.  
**Rujukan.** COBIT 2019.

**23) Organization–Performs–Audit**

Relasi ini menegaskan **Audit** dilaksanakan/dikoordinasi oleh **Organization** (aktor). Ontologis: **Organization** = agent; **Audit** = event.  
**Kardinalitas.** Organization ↔ Audit = **1** ↔ **0..**\*; dari Audit ke Organization **1..1** (akuntabilitas tunggal).  
**OCL.** context MetadataAudit inv: self.performedBy->size() = 1  
**Contoh.** Unit Kepatuhan melaksanakan “Audit Kepatuhan Metadata 2024”.  
**Rujukan.** ISO/IEC 27001 (9.2); ISO 19011; COBIT 2019 (MEA).

**24) Organization–Governs–Lifecycle**

Relasi ini menyatakan **Organization** mengatur pelaksanaan fase **Lifecycle**. Ontologis: **Organization** sebagai otoritas governance; **LifecyclePhase** tunduk pada wewenang tersebut.  
**Kardinalitas.** Organization ↔ LifecyclePhase = **1** ↔ **0..**\*; dari LifecyclePhase ke Organization **1..1**.  
**OCL.** context LifecyclePhase inv: self.governedBy->size() = 1  
**Contoh.** Kantor Data Governance mengatur fase “Publikasi” & “Dekomisioning”.  
**Rujukan.** ISO/IEC 11179; DAMA-DMBOK2; ISO/IEC 38500.

**25) Goal–Guides–Strategy**

Relasi ini menegaskan **Governance Goal** mengarahkan **Metadata Strategy**. Ontologis: *goals cascade* mentransformasikan kebutuhan pemangku kepentingan ke arah strategis yang dapat dieksekusi.  
**Kardinalitas.** Goal ↔ Strategy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataStrategy inv: self.goals->size()>=1  
**Contoh.** Goal “Interoperabilitas” → Strategy “Enterprise Data Standardization”.  
**Rujukan.** COBIT 2019 (Goals Cascade); ISO/IEC 38500.

**26) Strategy–Guides–Policy**

Relasi ini menyatakan **Strategy** mengarahkan **Policy** agar kebijakan konsisten dengan arah strategis organisasi (fungsi *direct* dalam governance).  
**Kardinalitas.** Strategy ↔ Policy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.strategy->notEmpty()  
**Contoh.** Strategy “Data Quality First” → Policy “Completeness ≥95%”.  
**Rujukan.** ISO/IEC 38500 (direct); COBIT 2019 (APO01/APO02); DAMA-DMBOK2.

SAMPAI SINI

=== nambah OCL

menyisipkan **label kardinalitas** (di dekat sumber & target) dan **label OCL ringkas** (di titik tengah tiap edge) langsung pada diagram.

**Download – Metamodel\_26\_labeled\_OCL\_4clusters.drawio**

Detail:

* **4 klaster warna** sesuai pembagian Anda (Policy Control; Organizational Accountability; Assurance & Risk Control; Governance Execution).
* **26 relasi** dengan label Inggris, **cardinality** di tepi edge (mis. 1..\* / 0..\*), dan **OCL** singkat (mis. OCL: context MetadataPolicy inv: self.standards->size()>=1) diletakkan di sekitar tengah garis agar tidak menumpuk pada node.
* Format .drawio kompatibel (atribut minimal, karakter OCL di-escape).

Kalau Anda ingin:

* ukuran/posisi label OCL diperkecil/diubah,
* menampilkan **hanya OCL untuk relasi tertentu** (biar tidak ramai),
* atau menambahkan **ikon klaster/legenda**,

tinggal bilang—saya buatkan variannya segera.

Yang lama (ga dipakai):

**Diagram .drawio (label Inggris saja)**  
**Download – Metamodel\_24\_english.drawio**  
(Semua edge sudah berlabel English: *Defines, Ensures, Guides, Constrains, Requires, Verified by, Assigns, Has, Oversees, Authorizes, Resolves, Validates, Reports, Impacts, Monitored by, Relates to, Tracked by, Measures, Performs, Governs*.)

**Katalog Relasi 24**

**Download – Metamodel\_24\_labeled\_4clusters.drawio**

**Empat klaster & isi**

* **Strategy & Assurance** (hijau): *MetadataGovernanceGoal, KPIIndicator, MetadataAudit, MetadataRisk*
* **Policy Framework** (biru): *ComplianceSource, MetadataPolicy, MetadataStandard, MetadataRule, MetadataAccessControl*
* **Organization & Quality** (oranye): *MetadataOrganization, MetadataStewardship, MetadataOwner, OwnerRole, MetadataQuality*
* **Lifecycle & Change** (biru muda): *LifecyclePhase, MetadataChangeRequest, MetadataIssueLog*

Semua relasi tetap berlabel **Inggris** dan kardinalitas ditempel di sisi sumber/target (mis. **1..**\* di dekat node sumber, **0..***/****1..*** di dekat node target).

| **#** | **Relation Name (EN)** | **Nama Relasi (IDN)** | **Tipe** | **Sumber→Target** | **Kardinalitas** | **Atribut Relasi (opsional)** | **OCL (ringkas)** | **Justifikasi Ontologis** | **Contoh Instansiasi** | **Rujukan** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Policy-Defines-Standard | Kebijakan-Mendefinisikan-Standar | Association | MetadataPolicy → MetadataStandard | 1..\* ↔ 1..\* | basis:string, effectiveDate:date | context MetadataPolicy inv: self.standards->size()>=1 | Normatif: kebijakan menurunkan standar teknis; bukan hierarki kelas. | “Kebijakan Metadata Nasional” → ISO/IEC 11179 | DAMA-DMBOK2; ISO/IEC 11179 |
| 2 | Policy-Defines-Rule | Kebijakan-Mendefinisikan-Aturan | Association | MetadataPolicy → MetadataRule | 1..\* ↔ 1..\* | rationale:string | context MetadataPolicy inv: self.rules->size()>=1 | Aturan operasional adalah turunan konkrit dari kebijakan. | Policy “Penamaan” → Rule “≤50 char” | DAMA-DMBOK2 |
| 3 | Policy-Ensures-Quality | Kebijakan-Memastikan-Kualitas | Association | MetadataPolicy → MetadataQuality | 1..\* ↔ 0..\* | target:decimal, operator:{≥,≤,=}, unit:string | context MetadataPolicy inv: self.qualityTargets->notEmpty() | Kebijakan menetapkan target kualitas (accuracy, completeness, dsb.). | Policy Kualitas → target completeness ≥95% | DAMA-DMBOK2 ch.13 |
| 4 | Goal-Guides-Policy | Tujuan-Mengarahkan-Kebijakan | Association | MetadataGovernanceGoal → MetadataPolicy | 1 ↔ 1..\* |  | context MetadataPolicy inv: self.goal->notEmpty() | Tujuan tata kelola mengarahkan pembentukan kebijakan. | Goal interoperabilitas → kebijakan standardisasi | ISO/IEC 38500 |
| 5 | Compliance-Constrains-Policy | SumberKepatuhan-Membatasi-Kebijakan | Association | ComplianceSource → MetadataPolicy | 1..\* ↔ 1..\* | clause:string | context MetadataPolicy inv: self.complianceSources->size()>=1 | Regulasi/standar eksternal membatasi/menjadi acuan kebijakan. | GDPR → Kebijakan Retensi | GDPR; ISO 27001 |
| 6 | Compliance-Constrains-Rule | SumberKepatuhan-Membatasi-Aturan | Association | ComplianceSource → MetadataRule | 1..\* ↔ 1..\* | clause:string | context MetadataRule inv: self.complianceSources->size()>=1 | Aturan spesifik sering bersumber dari regulasi. | GDPR → Rule penyimpanan data | GDPR |
| 7 | Rule-Requires-AccessControl | Aturan-Memerlukan-KontrolAkses | Association | MetadataRule → MetadataAccessControl | 1..\* ↔ 1..\* | controlType:string | context MetadataRule inv: self.accessControls->size()>=1 | Penerapan rule membutuhkan kontrol akses berdasar peran/kelas data. | Rule akses-atribut → ACL/ABAC | ISO/IEC 27001 |
| 8 | Rule-VerifiedBy-Audit | Aturan-DiverifikasiOleh-Audit | Association | MetadataRule → MetadataAudit | 1..\* ↔ 1..\* | auditType:string, sample:int | context MetadataRule inv: self.audits->size()>=1 | Kepatuhan rule diverifikasi melalui audit. | Audit 2024 memeriksa Rule penamaan | COBIT 2019; ISO 27001 |
| 9 | Organization-Assigns-Stewardship | Organisasi-Menetapkan-Stewardship | Association (dapat jadi AssociationClass) | MetadataOrganization → MetadataStewardship | 1 ↔ 1..\* | mandateDoc:string, appointmentDate:date, scope:string | context MetadataOrganization inv: self.stewardships->size()>=1 | Mandat stewardship merupakan keputusan organisasi (aktor). | BPS menetapkan Stewardship Katalog Metadata | DAMA-DMBOK2 |
| 10 | Stewardship-Has-Owner | Stewardship-Memiliki-Pemilik | Association | MetadataStewardship → MetadataOwner | 1..\* ↔ 1..1 | startDate:date | context MetadataStewardship inv: self.owner->size()=1 | Setiap stewardship memiliki pemilik tunggal yang akuntabel. | Stewardship A → Owner “Dir. Statistik” | Peraturan internal |
| 11 | Owner-Has-Role | Pemilik-Memiliki-Peran | Association | MetadataOwner → OwnerRole | 1 ↔ 1..\* | roleType:{DataOwner, Custodian, Steward} | context MetadataOwner inv: self.roles->notEmpty() | Seorang owner menjalankan satu atau lebih peran formal. | Owner X → Role Data Custodian | ISO/IEC 38500 |
| 12 | Owner-Oversees-Quality | Pemilik-Mengawasi-Kualitas | Association | MetadataOwner → MetadataQuality | 1 ↔ 1..\* |  | context MetadataOwner inv: self.qualityTargets->notEmpty() | Akuntabilitas kualitas ditetapkan kepada owner. | Owner menetapkan target completeness | DAMA-DMBOK2 |
| 13 | Owner-Authorizes-Change | Pemilik-Mengesahkan-Perubahan | Association | MetadataOwner → MetadataChangeRequest | 1 ↔ 0..\* | decision:{approve,reject}, decisionDate:date | context MetadataChangeRequest inv: self.authorizedBy->size()<=1 | Perubahan definisi harus disahkan owner. | Owner menyetujui perubahan definisi | COBIT 2019 |
| 14 | Owner-Resolves-Issue | Pemilik-Menyelesaikan-Isu | Association | MetadataOwner → MetadataIssueLog | 1 ↔ 0..\* | resolutionDate:date, severity:{low,med,high} | context MetadataIssueLog inv: self.resolvedBy->size()<=1 | Isu terkait metadata ditangani oleh owner yang relevan. | Owner menyelesaikan isu validasi | DAMA-DMBOK2 |
| 15 | Audit-Validates-Policy | Audit-Memvalidasi-Kebijakan | Association | MetadataAudit → MetadataPolicy | 1..\* ↔ 1..\* | auditPeriod:dateRange | context MetadataAudit inv: self.policies->notEmpty() | Audit menilai kepatuhan kebijakan. | Audit Kepatuhan Kebijakan Metadata | ISO 27001 |
| 16 | Audit-Reports-KPI | Audit-Melaporkan-KPI | Association | MetadataAudit → KPIIndicator | 1..\* ↔ 0..\* | value:decimal, unit:string, period:dateRange | context MetadataAudit inv: self.kpis->notEmpty() | Audit menghasilkan indikator kinerja untuk pelaporan governance. | Audit 2024 → KPI kepatuhan 90% | COBIT 2019 |
| 17 | Risk-Impacts-Quality | Risiko-BerdampakPada-Kualitas | Association | MetadataRisk → MetadataQuality | 1..\* ↔ 1..\* | impact:{H/M/L} | context MetadataRisk inv: self.affectedQuality->notEmpty() | Risiko memengaruhi dimensi kualitas. | Risiko integritas → konsistensi metadata | ISO 27001 |
| 18 | Risk-MonitoredBy-Audit | Risiko-DimonitorOleh-Audit | Association | MetadataRisk → MetadataAudit | 1..\* ↔ 1..\* |  | context MetadataRisk inv: self.monitoringAudits->notEmpty() | Pengendalian risiko dimonitor melalui audit. | Audit mitigasi risiko metadata | DAMA-DMBOK2 |
| 19 | ChangeRequest-Relates-Issue | PermintaanPerubahan-TerkaitDengan-Isu | Association (dapat jadi AssociationClass) | MetadataChangeRequest → MetadataIssueLog | 1..\* ↔ 0..\* | relationType:{originatesFrom,duplicates,blocks} | context MetadataChangeRequest inv: self.issues->notEmpty() | Permintaan perubahan biasanya muncul dari isu-isu terdokumentasi. | Isu validasi → CR #123 | DAMA-DMBOK2 |
| 20 | Lifecycle-Impacts-Change | SiklusHidup-BerdampakPada-Perubahan | Association | LifecyclePhase → MetadataChangeRequest | 1..\* ↔ 0..\* | phase:{design,publication,deprecation,decommission} | context LifecyclePhase inv: self.changeRequests->notEmpty() | Fase siklus hidup menentukan kebutuhan perubahan. | Dekomisioning → CR penghapusan metadata | ISO/IEC 11179 |
| 21 | Lifecycle-TrackedBy-Audit | SiklusHidup-DilacakOleh-Audit | Association | LifecyclePhase → MetadataAudit | 1..\* ↔ 0..\* | checkpoint:string | context LifecyclePhase inv: self.audits->notEmpty() | Setiap fase lifecycle memiliki checkpoint audit. | Audit fase publikasi | DAMA-DMBOK2 |
| 22 | KPI-Measures-Goal | KPI-Mengukur-Tujuan | Association | KPIIndicator → MetadataGovernanceGoal | 1..\* ↔ 1..\* | target:decimal, threshold:decimal, direction:{↑,↓} | context KPIIndicator inv: self.goal->notEmpty() | KPI mengukur ketercapaian goal tata kelola. | KPI kepatuhan → Goal kualitas | COBIT 2019 |
| 23 | Organization-Performs-Audit | Organisasi-Melaksanakan-Audit | Association | MetadataOrganization → MetadataAudit | 1 ↔ 0..\* | unit:string | context MetadataOrganization inv: self.audits->notEmpty() | Audit dilaksanakan/dikoordinasi oleh organisasi (aktor). | Unit Kepatuhan → Audit 2024 | ISO 27001; COBIT 2019 |
| 24 | Organization-Governs-Lifecycle | Organisasi-Mengatur-SiklusHidup | Association | MetadataOrganization → LifecyclePhase | 1 ↔ 0..\* | policyRef:string | context MetadataOrganization inv: self.lifecyclePhases->notEmpty() | Organisasi mengatur pelaksanaan fase siklus hidup metadata. | Divisi Data → fase publikasi/dekomisioning | ISO/IEC 11179; DAMA-DMBOK2 |

Penjelasan:  
yang baru:

**1) Policy–Defines–Standard**

**Relasi ini** menegaskan bahwa setiap kebijakan metadata harus diwujudkan dalam bentuk standar teknis. **Secara ontologis**, *Policy* berada pada tingkat prinsip/aturan normatif, sedangkan *Standard* mengoperasionalkan prinsip tersebut (aturan implementatif yang dapat diaudit).  
**Kardinalitas (disarankan).** Policy ↔ Standard = **1..**\* ↔ **1..**\* (setiap kebijakan menghasilkan ≥1 standar; setiap standar merujuk ≥1 kebijakan).  
**OCL (ringkas).** context MetadataPolicy inv: self.standards->size()>=1  
**Contoh.** Kebijakan Metadata Nasional menurunkan Standar ISO/IEC 11179 untuk penamaan elemen data.  
**Rujukan.** DAMA-DMBOK2; ISO/IEC 11179.

**2) Policy–Defines–Rule**

**Relasi ini** menyatakan bahwa aturan operasional (*Rule*) adalah turunan konkrit dari kebijakan. **Secara ontologis**, *Rule* adalah constraint/prosedur yang memampukan penilaian kepatuhan terhadap *Policy*.  
**Kardinalitas.** Policy ↔ Rule = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.rules->size()>=1  
**Contoh.** Kebijakan “Penamaan” → Rule “Panjang nama atribut ≤ 50 karakter”.  
**Rujukan.** DAMA-DMBOK2.

**3) Policy–Ensures–Quality**

**Relasi ini** menyatakan kebijakan menetapkan target pada dimensi kualitas metadata (*QualityDimension*). **Ontologis**, *Policy* men-setting nilai sasaran (mis. akurasi, kelengkapan) yang dapat diuji.  
**Kardinalitas.** Policy ↔ Quality = **1..**\* ↔ **0..**\* (kebijakan dapat menetapkan banyak target; satu dimensi dapat—atau tidak—ditargetkan).  
**OCL.** context MetadataPolicy inv: self.qualityTargets->notEmpty()  
**Contoh.** Kebijakan Kualitas → *completeness* ≥ 95%.  
**Rujukan.** DAMA-DMBOK2 (Bab Kualitas Data).

**4) Goal–Guides–Policy**

**Relasi ini** menyatakan tujuan tata kelola (*GovernanceGoal*) mengarahkan pembentukan kebijakan. **Ontologis**, *Goal* berada di level teleologis (arah), *Policy* di level normatif (cara).  
**Kardinalitas.** Goal ↔ Policy = **1** ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.goal->notEmpty()  
**Contoh.** *Goal* interoperabilitas → kebijakan standardisasi metadata.  
**Rujukan.** ISO/IEC 38500 (prinsip tata kelola TI).

**5) Compliance–Constrains–Policy**

**Relasi ini** menyatakan sumber kepatuhan (regulasi/standar) **membatasi/mengarahkan** kebijakan. **Ontologis**, *ComplianceSource* adalah sumber normatif eksternal terhadap *Policy*.  
**Kardinalitas.** Compliance ↔ Policy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataPolicy inv: self.complianceSources->size()>=1  
**Contoh.** GDPR → Kebijakan Retensi.  
**Rujukan.** GDPR; ISO/IEC 27001.

**6) Compliance–Constrains–Rule**

**Relasi ini** menyatakan *Rule* spesifik sering langsung diturunkan dari regulasi. **Ontologis**, *ComplianceSource* adalah pembatas/pemberi kewajiban eksternal bagi *Rule*.  
**Kardinalitas.** Compliance ↔ Rule = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.complianceSources->size()>=1  
**Contoh.** GDPR → aturan “retention maximum N tahun”.  
**Rujukan.** GDPR.

**7) Rule–Requires–AccessControl**

**Relasi ini** menyatakan penegakan *Rule* membutuhkan kontrol akses (*AccessControl*). **Ontologis**, *AccessControl* adalah mekanisme pelaksana constraint *Rule*.  
**Kardinalitas.** Rule ↔ AccessControl = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.accessControls->size()>=1  
**Contoh.** Rule akses atribut sensitif → ACL/ABAC berdasar *OwnerRole*.  
**Rujukan.** ISO/IEC 27001.

**8) Rule–VerifiedBy–Audit**

**Relasi ini** menyatakan kepatuhan *Rule* diverifikasi melalui *Audit*. **Ontologis**, *Audit* adalah event evaluatif terhadap penerapan *Rule*.  
**Kardinalitas.** Rule ↔ Audit = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRule inv: self.audits->size()>=1  
**Contoh.** Audit 2024 memeriksa rule penamaan atribut.  
**Rujukan.** COBIT 2019; ISO/IEC 27001.

**9) Organization–Assigns–Stewardship**

**Relasi ini** menegaskan mandat stewardship ditetapkan oleh organisasi sebagai aktor institusional. **Ontologis**, *Organization* adalah *agent*, *Stewardship* adalah peran/proses tata kelola yang diberi mandat.  
**Kardinalitas.** Organization ↔ Stewardship = **1** ↔ **1..**\*.  
**OCL.** context MetadataOrganization inv: self.stewardships->size()>=1  
**Contoh.** BPS menetapkan *Stewardship* Katalog Metadata.  
**Rujukan.** DAMA-DMBOK2.

**10) Stewardship–Has–Owner**

**Relasi ini** menyatakan tiap stewardship memiliki satu pemilik akuntabel. **Ontologis**, *Owner* adalah entitas dengan *responsibility* final atas domain stewardship.  
**Kardinalitas.** Stewardship ↔ Owner = **1..**\* ↔ **1..1**.  
**OCL.** context MetadataStewardship inv: self.owner->size()=1  
**Contoh.** Stewardship A → Owner “Direktur Statistik”.  
**Rujukan.** Kebijakan internal; praktik RACI.

**11) Owner–Has–Role**

**Relasi ini** menyatakan seorang *Owner* menjalankan satu atau lebih *OwnerRole* (mis. Data Owner, Custodian, Steward). **Ontologis**, *Role* mengikat hak/otoritas/fungsi.  
**Kardinalitas.** Owner ↔ OwnerRole = **1** ↔ **1..**\*.  
**OCL.** context MetadataOwner inv: self.roles->notEmpty()  
**Contoh.** Owner X → *Role* Data Custodian.  
**Rujukan.** ISO/IEC 38500.

**12) Owner–Oversees–Quality**

**Relasi ini** menegaskan akuntabilitas kualitas metadata pada *Owner*. **Ontologis**, *Owner* adalah *accountability bearer* atas dimensi kualitas.  
**Kardinalitas.** Owner ↔ Quality = **1** ↔ **1..**\*.  
**OCL.** context MetadataOwner inv: self.qualityTargets->notEmpty()  
**Contoh.** Owner menetapkan target *completeness*.  
**Rujukan.** DAMA-DMBOK2.

**13) Owner–Authorizes–Change**

**Relasi ini** menyatakan perubahan definisi metadata (CR) harus disahkan *Owner*. **Ontologis**, *Owner* adalah *authority* yang memberi legitimasi pada *ChangeRequest*.  
**Kardinalitas.** Owner ↔ ChangeRequest = **1** ↔ **0..**\*; dari CR ke Owner **0..1** (satu pemberi otorisasi utama).  
**OCL.** context MetadataChangeRequest inv: self.authorizedBy->size()<=1  
**Contoh.** Owner menyetujui CR untuk memperbarui definisi elemen.  
**Rujukan.** COBIT 2019 (manajemen perubahan).

**14) Owner–Resolves–Issue**

**Relasi ini** menegaskan *IssueLog* metadata harus ditangani oleh *Owner* terkait. **Ontologis**, *Owner* sebagai penanggung jawab *issue lifecycle*.  
**Kardinalitas.** Owner ↔ IssueLog = **1** ↔ **0..**\*; dari Issue ke Owner **0..1**.  
**OCL.** context MetadataIssueLog inv: self.resolvedBy->size()<=1  
**Contoh.** Owner menyelesaikan isu validasi definisi elemen.  
**Rujukan.** DAMA-DMBOK2.

**15) Audit–Validates–Policy**

**Relasi ini** menyatakan audit menilai kepatuhan terhadap *Policy*. **Ontologis**, *Audit* adalah event verifikasi terhadap aturan normatif *Policy*.  
**Kardinalitas.** Audit ↔ Policy = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataAudit inv: self.policies->notEmpty()  
**Contoh.** Audit Kepatuhan Kebijakan Metadata.  
**Rujukan.** ISO/IEC 27001; ISO 19011.

**16) Audit–Reports–KPI**

**Relasi ini** menyatakan audit menghasilkan indikator kinerja (*KPIIndicator*) untuk pelaporan governance. **Ontologis**, *KPI* adalah artefak evaluatif dari *Audit*.  
**Kardinalitas.** Audit ↔ KPI = **1..**\* ↔ **0..**\*.  
**OCL.** context MetadataAudit inv: self.kpis->notEmpty()  
**Contoh.** Audit 2024 → KPI kepatuhan 90%.  
**Rujukan.** COBIT 2019.

**17) Risk–Impacts–Quality**

**Relasi ini** menyatakan *Risk* berdampak pada dimensi kualitas metadata. **Ontologis**, *Risk* sebagai potensi kejadian memengaruhi properti *Quality*.  
**Kardinalitas.** Risk ↔ Quality = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRisk inv: self.affectedQuality->notEmpty()  
**Contoh.** Risiko integritas → konsistensi metadata.  
**Rujukan.** ISO/IEC 27001 (manajemen risiko).

**18) Risk–MonitoredBy–Audit**

**Relasi ini** menyatakan pengendalian risiko dimonitor melalui *Audit*. **Ontologis**, *Audit* adalah mekanisme penilaian efektivitas kontrol atas *Risk*.  
**Kardinalitas.** Risk ↔ Audit = **1..**\* ↔ **1..**\*.  
**OCL.** context MetadataRisk inv: self.monitoringAudits->notEmpty()  
**Contoh.** Audit mitigasi risiko metadata tahunan.  
**Rujukan.** DAMA-DMBOK2; ISO/IEC 27001.

**19) ChangeRequest–Relates–Issue**

**Relasi ini** menyatakan permintaan perubahan biasanya muncul dari isu terdokumentasi. **Ontologis**, *Issue* adalah sebab/justifikasi yang *menautkan* ke *ChangeRequest*.  
**Kardinalitas.** ChangeRequest ↔ IssueLog = **1..**\* ↔ **0..**\* (CR boleh mereferensikan ≥1 isu; satu isu bisa 0 atau banyak CR).  
**OCL.** context MetadataChangeRequest inv: self.issues->notEmpty()  
**Contoh.** Isu validasi → CR #123.  
**Rujukan.** DAMA-DMBOK2.

**20) Lifecycle–Impacts–Change**

**Relasi ini** menyatakan fase siklus hidup memicu kebutuhan perubahan. **Ontologis**, *LifecyclePhase* adalah keadaan/proses institusional yang memunculkan *ChangeRequest*.  
**Kardinalitas.** LifecyclePhase ↔ ChangeRequest = **1..**\* ↔ **0..**\*.  
**OCL.** context LifecyclePhase inv: self.changeRequests->notEmpty()  
**Contoh.** Fase dekomisioning → CR penghapusan metadata.  
**Rujukan.** ISO/IEC 11179.

**21) Lifecycle–TrackedBy–Audit**

**Relasi ini** menyatakan tiap fase lifecycle memiliki checkpoint audit. **Ontologis**, *Audit* mendokumentasi/memverifikasi transisi fase *Lifecycle*.  
**Kardinalitas.** LifecyclePhase ↔ Audit = **1..**\* ↔ **0..**\*.  
**OCL.** context LifecyclePhase inv: self.audits->notEmpty()  
**Contoh.** Audit pada fase publikasi metadata.  
**Rujukan.** DAMA-DMBOK2.

**22) KPI–Measures–Goal**

**Relasi ini** menyatakan KPI mengukur ketercapaian *GovernanceGoal*. **Ontologis**, *KPIIndicator* adalah predikat kuantitatif atas *Goal*.  
**Kardinalitas.** KPI ↔ Goal = **1..**\* ↔ **1..**\*.  
**OCL.** context KPIIndicator inv: self.goal->notEmpty()  
**Contoh.** KPI “Kepatuhan 90%” → Goal “Kualitas Metadata”.  
**Rujukan.** COBIT 2019.

**23) Organization–Performs–Audit**

**Relasi ini** menegaskan audit dilaksanakan/dikoordinasi oleh organisasi (aktor). **Ontologis**, *Organization* = *agent*; *Audit* = *event*.  
**Kardinalitas.** Organization ↔ Audit = **1** ↔ **0..**\*; dari Audit ke Organization **1..1** (akuntabilitas tunggal).  
**OCL.** context MetadataAudit inv: self.performedBy->size() = 1  
**Contoh.** Unit Kepatuhan melaksanakan “Audit Kepatuhan Metadata 2024”.  
**Rujukan.** ISO/IEC 27001 (9.2); ISO 19011; COBIT 2019 (MEA).

**24) Organization–Governs–Lifecycle**

**Relasi ini** menyatakan organisasi mengatur pelaksanaan fase siklus hidup metadata. **Ontologis**, *Organization* sebagai otoritas/penentu kebijakan; *LifecyclePhase* sebagai state/proses institusional yang tunduk pada governance.  
**Kardinalitas.** Organization ↔ LifecyclePhase = **1** ↔ **0..**\*; dari LifecyclePhase ke Organization **1..1** (RACI jelas).  
**OCL.** context LifecyclePhase inv: self.governedBy->size() = 1  
**Contoh.** Kantor Data Governance mengatur fase “Publikasi” dan “Dekomisioning” (syarat CR & bukti audit).  
**Rujukan.** ISO/IEC 11179; DAMA-DMBOK2; ISO/IEC 38500.

yang lama:

**Relasi Metamodel Metadata Governance**

1. **Policy–Defines–Standard**  
   Relasi ini menegaskan bahwa setiap kebijakan metadata harus diwujudkan dalam bentuk standar teknis. Secara ontologis, kebijakan berada pada level prinsip, sedangkan standar mengoperasionalkan prinsip tersebut. Contoh: *Kebijakan Metadata Nasional* menurunkan *Standar ISO/IEC 11179* sebagai rujukan penamaan elemen data. Hal ini sesuai dengan DAMA-DMBOK2 (2024) dan ISO/IEC 11179.
2. **Policy–Defines–Rule**  
   Relasi ini menunjukkan bahwa kebijakan metadata melahirkan aturan-aturan rinci. Ontologinya adalah derivasi: kebijakan memandu pembentukan aturan. Misalnya, *Kebijakan interoperabilitas* menurunkan *aturan penamaan field ≤ 50 karakter*.
3. **Policy–Ensures–Quality**  
   Kebijakan harus mencakup dimensi kualitas metadata. Secara ontologis, kualitas menjadi tujuan normatif yang diikat kebijakan. Contoh: kebijakan menyatakan metadata harus memiliki *akurasi ≥ 95%*. Sumber: DAMA-DMBOK2 Bab 13.
4. **Policy–GuidedBy–Goal**  
   Kebijakan metadata tidak berdiri sendiri, melainkan diarahkan oleh tujuan tata kelola metadata. Ontologinya adalah relasi tujuan–instrumen. Contoh: *Goal interoperabilitas* menghasilkan *kebijakan standar metadata nasional*. Rujukan: ISO 38500.
5. **Policy–CompliesWith–Source**  
   Kebijakan harus patuh pada sumber eksternal (hukum atau regulasi). Ontologinya adalah relasi kepatuhan. Contoh: *GDPR* → *Kebijakan retensi metadata*.
6. **Rule–Validates–Compliance**  
   Aturan metadata harus memvalidasi kepatuhan pada regulasi. Ontologinya adalah aturan sebagai mekanisme implementasi. Contoh: *GDPR* → *Rule penyimpanan maksimum 5 tahun*.
7. **Rule–GovernedBy–AccessControl**  
   Aturan metadata mengatur mekanisme kontrol akses. Ontologinya adalah pembatasan perilaku oleh rule. Contoh: aturan hak akses hanya untuk *data steward*. Rujukan: ISO/IEC 27001.
8. **Rule–MeasuredBy–Audit**  
   Aturan harus diuji melalui audit. Ontologinya adalah verifikasi eksternal terhadap norma. Contoh: audit kepatuhan *aturan penamaan*. Rujukan: COBIT 2019.
9. **Organization–Assigns–Stewardship**  
   Organisasi memiliki kewajiban formal untuk menunjuk stewardship. Ontologinya adalah mandat kelembagaan. Contoh: *BPS* menunjuk *Data Steward Statistik*. Rujukan: DAMA-DMBOK2.
10. **Stewardship–Has–Owner**  
    Setiap stewardship harus memiliki pemilik formal. Ontologinya adalah prinsip akuntabilitas. Contoh: stewardship metadata berada di bawah *Dirjen Statistik*.
11. **Owner–Has–Role**  
    Pemilik metadata harus dijabarkan ke dalam satu atau lebih peran. Ontologinya adalah abstraksi peran dalam ISO 38500. Contoh: *Data Custodian* atau *Data Producer*.
12. **Owner–Oversees–Quality**  
    Pemilik bertanggung jawab atas kualitas metadata. Ontologinya adalah akuntabilitas atas outcome. Contoh: pemilik menetapkan *target completeness ≥ 98%*.
13. **Owner–Authorizes–Change**  
    Setiap perubahan metadata harus disahkan oleh pemilik. Ontologinya adalah relasi otorisasi. Contoh: perubahan definisi atribut hanya dapat disetujui oleh pemilik.
14. **Owner–Resolves–Issue**  
    Isu metadata wajib diselesaikan oleh pemilik. Ontologinya adalah pertanggungjawaban. Contoh: pemilik menyelesaikan isu *duplikasi metadata*.
15. **Audit–Validates–Policy**  
    Audit menguji kepatuhan terhadap kebijakan metadata. Ontologinya adalah kontrol eksternal. Contoh: audit kepatuhan *Kebijakan Metadata Nasional*.
16. **Audit–Reports–KPI**  
    Audit menghasilkan indikator kinerja yang dapat diukur. Ontologinya adalah hasil observasi → ukuran. Contoh: *Audit 2024* menghasilkan KPI kepatuhan 90%.
17. **Risk–Impacts–Quality**  
    Risiko metadata berdampak langsung pada kualitas. Ontologinya adalah kausalitas. Contoh: risiko integritas → menurunkan konsistensi metadata.
18. **Risk–MonitoredBy–Audit**  
    Risiko metadata harus dipantau oleh audit. Ontologinya adalah pengawasan risiko. Contoh: audit menguji mitigasi risiko *akses ilegal*.
19. **ChangeRequest–Relates–Issue**  
    Permintaan perubahan metadata biasanya muncul dari isu. Ontologinya adalah relasi sebab-akibat. Contoh: isu *field kosong* → permintaan perubahan definisi.
20. **Lifecycle–Impacts–Change**  
    Tahap siklus hidup metadata menentukan kebutuhan perubahan. Ontologinya adalah relasi fase–aksi. Contoh: fase *dekomisioning* → request penghapusan metadata.
21. **Lifecycle–TrackedBy–Audit**  
    Audit digunakan untuk meninjau kepatuhan di setiap fase lifecycle. Ontologinya adalah monitoring fase. Contoh: audit pada fase publikasi metadata.
22. **KPI–Measures–Goal**  
    Indikator kinerja (KPI) mengukur ketercapaian tujuan metadata governance. Ontologinya adalah relasi evaluasi. Contoh: KPI kepatuhan → goal *kualitas metadata*.

**🔹 Sintesis Relasi Metamodel Metadata Governance**

Dua puluh dua relasi yang dirancang dalam metamodel metadata governance membentuk sebuah **ekosistem tata kelola** yang bersifat utuh dan saling terhubung. Relasi-relasi tersebut tidak hanya bersifat teknis, melainkan mengandung dimensi **normatif, organisasional, kontrol, dan evaluatif**.

Dari sisi **normatif**, relasi antara *Policy–Standard–Rule–ComplianceSource* menegaskan bahwa tata kelola metadata selalu berpijak pada kebijakan yang sahih, diturunkan ke dalam standar, dioperasionalkan melalui aturan, dan berakar pada regulasi formal. Hal ini memastikan bahwa metadata governance memiliki legitimasi hukum dan arah strategis yang jelas.

Dari sisi **organisasional**, relasi *Organization–Stewardship–Owner–Role* menekankan prinsip **akuntabilitas dan tanggung jawab manusia** dalam tata kelola. Ontologi yang dibangun menegaskan bahwa setiap steward harus memiliki pemilik, dan setiap pemilik harus menjalankan peran yang spesifik. Dengan demikian, tata kelola metadata tidak sekadar mekanisme teknis, melainkan struktur sosial yang mengikat organisasi.

Dari sisi **kontrol dan risiko**, relasi *Rule–Audit–AccessControl–Risk* menunjukkan bagaimana aturan diterapkan, diuji, dan dikendalikan. Audit berperan sebagai mekanisme verifikasi independen, sementara risiko dikelola secara proaktif melalui hubungan dengan kualitas dan audit. Relasi ini merepresentasikan dimensi **assurance & risk control** dalam governance.

Dari sisi **evaluasi dan perbaikan berkelanjutan**, relasi *ChangeRequest–Issue–Lifecycle–KPI–Goal* membentuk siklus reflektif. Isu memunculkan permintaan perubahan, perubahan dipandu oleh fase siklus hidup, dan seluruhnya dievaluasi melalui audit serta diukur dengan indikator kinerja terhadap tujuan. Hal ini memastikan metadata governance bersifat **adaptif dan evolutif**.

Dengan demikian, keseluruhan 22 relasi dalam metamodel ini menggambarkan **rantai kendali yang utuh**: mulai dari tujuan strategis → kebijakan → standar → aturan → kontrol organisasi → audit → evaluasi → perubahan. Keterhubungan ini bukan sekadar visualisasi, melainkan **kerangka konseptual ontologis** yang memungkinkan organisasi memahami, merancang, dan mengimplementasikan tata kelola metadata secara konsisten, terukur, dan dapat dipertanggungjawabkan.

📌 Narasi penutup ini bisa Anda gunakan sebagai jembatan sebelum masuk ke bagian validasi metamodel (misalnya di Subbab 4.4.3).

Apakah Anda ingin saya lengkapi juga dengan **diagram sintesis relasi** (misalnya peta hubungan antar klaster Policy Control – Organizational Accountability – Assurance & Risk – Execution) agar penguji lebih mudah memahami pola keterhubungan 22 relasi ini?

**🔹 Narasi Diagram Sintesis Relasi**

Diagram peta relasi metamodel metadata governance dibangun dalam empat klaster:

1. **Policy Control**
   * Elemen: *MetadataGovernanceGoal, MetadataPolicy, MetadataStandard, MetadataRule, ComplianceSource, MetadataQuality*
   * Relasi utama: *Goal → Policy → Standard & Rule*, *Policy → ComplianceSource*, *Policy → Quality*.
   * Makna: memastikan kebijakan metadata diarahkan oleh tujuan strategis dan tunduk pada regulasi formal.
2. **Organizational Accountability**
   * Elemen: *MetadataOrganization, MetadataStewardship, MetadataOwner, OwnerRole*
   * Relasi utama: *Organization → Stewardship → Owner → Role*.
   * Makna: menegaskan akuntabilitas formal dan struktur peran dalam governance.
3. **Assurance & Risk Control**
   * Elemen: *MetadataAudit, MetadataRisk, MetadataAccessControl, MetadataQuality*
   * Relasi utama: *Rule → AccessControl*, *Rule → Audit*, *Risk → Audit*, *Risk → Quality*, *Audit → KPI*.
   * Makna: menyediakan mekanisme pengawasan, mitigasi risiko, dan evaluasi kualitas metadata.
4. **Governance Execution**
   * Elemen: *MetadataChangeRequest, MetadataIssueLog, LifecyclePhase, KPIIndicator*
   * Relasi utama: *Issue → ChangeRequest*, *Lifecycle → ChangeRequest & Audit*, *KPI → Goal*.
   * Makna: mendukung siklus perbaikan berkelanjutan melalui manajemen isu, perubahan, dan pengukuran kinerja.

Metamodel\_Governance\_Clusters\_WithRelations.drawio

VERSI BAHASA INGGRIS

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Relation (EN (IDN))** | **Type** | **Source→Target** | **Cardinality** | **Relation Attributes (opt.)** | **OCL (brief)** | **Ontological Justification** | **Example Instantiation** | **References** |
| 1 | Policy-Defines-Standard **(Kebijakan-Mendefinisikan-Standar)** | Association | MetadataPolicy → MetadataStandard | 1..\* ↔ 1..\* | basis:string; effectiveDate:date | context MetadataPolicy inv: self.standards->size()>=1 | Policies normatively yield technical standards (not class hierarchy). | “National Metadata Policy” → ISO/IEC 11179 | DAMA-DMBOK2; ISO/IEC 11179 |
| 2 | Policy-Defines-Rule **(Kebijakan-Mendefinisikan-Aturan)** | Association | MetadataPolicy → MetadataRule | 1..\* ↔ 1..\* | rationale:string | context MetadataPolicy inv: self.rules->size()>=1 | Operational rules are concrete derivations of policies. | Policy “Naming” → Rule “≤ 50 chars” | DAMA-DMBOK2 |
| 3 | Policy-Ensures-Quality **(Kebijakan-Memastikan-Kualitas)** | Association | MetadataPolicy → MetadataQuality | 1..\* ↔ 0..\* | target:decimal; operator:{≥,≤,=}; unit:string | context MetadataPolicy inv: self.qualityTargets->notEmpty() | Policies set target values for quality dimensions. | Quality Policy → completeness ≥ 95% | DAMA-DMBOK2 ch.13 |
| 4 | Goal-Guides-Policy **(Tujuan-Mengarahkan-Kebijakan)** | Association | MetadataGovernanceGoal → MetadataPolicy | 1 ↔ 1..\* | — | context MetadataPolicy inv: self.goal->notEmpty() | Governance goals orient the formation of policies. | Interoperability goal → standardization policy | ISO/IEC 38500 |
| 5 | Compliance-Constrains-Policy **(SumberKepatuhan-Membatasi-Kebijakan)** | Association | ComplianceSource → MetadataPolicy | 1..\* ↔ 1..\* | clause:string | context MetadataPolicy inv: self.complianceSources->size()>=1 | External regulations bound and inform policies. | GDPR → Retention Policy | GDPR; ISO 27001 |
| 6 | Compliance-Constrains-Rule **(SumberKepatuhan-Membatasi-Aturan)** | Association | ComplianceSource → MetadataRule | 1..\* ↔ 1..\* | clause:string | context MetadataRule inv: self.complianceSources->size()>=1 | Specific rules often stem from regulations. | GDPR → Storage rule | GDPR |
| 7 | Rule-Requires-AccessControl **(Aturan-Memerlukan-KontrolAkses)** | Association | MetadataRule → MetadataAccessControl | 1..\* ↔ 1..\* | controlType:string | context MetadataRule inv: self.accessControls->size()>=1 | Rule enforcement needs role/data-class based controls. | Attribute-level rule → ACL/ABAC | ISO/IEC 27001 |
| 8 | Rule-VerifiedBy-Audit **(Aturan-DiverifikasiOleh-Audit)** | Association | MetadataRule → MetadataAudit | 1..\* ↔ 1..\* | auditType:string; sample:int | context MetadataRule inv: self.audits->size()>=1 | Rule compliance is checked via audit. | 2024 audit checks naming rule | COBIT 2019; ISO 27001 |
| 9 | Organization-Assigns-Stewardship **(Organisasi-Menetapkan-Stewardship)** | Association *(may be AssociationClass)* | MetadataOrganization → MetadataStewardship | 1 ↔ 1..\* | mandateDoc:string; appointmentDate:date; scope:string | context MetadataOrganization inv: self.stewardships->size()>=1 | Stewardship is an organizational mandate/act. | BPS appoints Metadata Catalog Stewardship | DAMA-DMBOK2 |
| 10 | Stewardship-Has-Owner **(Stewardship-Memiliki-Pemilik)** | Association | MetadataStewardship → MetadataOwner | 1..\* ↔ 1..1 | startDate:date | context MetadataStewardship inv: self.owner->size()=1 | Each stewardship has a single accountable owner. | Stewardship A → Owner “Director of Statistics” | Internal policy |
| 11 | Owner-Has-Role **(Pemilik-Memiliki-Peran)** | Association | MetadataOwner → OwnerRole | 1 ↔ 1..\* | roleType:{DataOwner,Custodian,Steward} | context MetadataOwner inv: self.roles->notEmpty() | An owner may play one or more formal roles. | Owner X → Data Custodian | ISO/IEC 38500 |
| 12 | Owner-Oversees-Quality **(Pemilik-Mengawasi-Kualitas)** | Association | MetadataOwner → MetadataQuality | 1 ↔ 1..\* | — | context MetadataOwner inv: self.qualityTargets->notEmpty() | Quality accountability is assigned to owners. | Owner sets completeness targets | DAMA-DMBOK2 |
| 13 | Owner-Authorizes-Change **(Pemilik-Mengesahkan-Perubahan)** | Association | MetadataOwner → MetadataChangeRequest | 1 ↔ 0..\* | decision:{approve,reject}; decisionDate:date | context MetadataChangeRequest inv: self.authorizedBy->size()<=1 | Definition changes must be authorized by owner. | Owner approves definition change | COBIT 2019 |
| 14 | Owner-Resolves-Issue **(Pemilik-Menyelesaikan-Isu)** | Association | MetadataOwner → MetadataIssueLog | 1 ↔ 0..\* | resolutionDate:date; severity:{low,med,high} | context MetadataIssueLog inv: self.resolvedBy->size()<=1 | Metadata issues are resolved by the relevant owner. | Owner resolves validation issue | DAMA-DMBOK2 |
| 15 | Audit-Validates-Policy **(Audit-Memvalidasi-Kebijakan)** | Association | MetadataAudit → MetadataPolicy | 1..\* ↔ 1..\* | auditPeriod:dateRange | context MetadataAudit inv: self.policies->notEmpty() | Audits validate policy conformance. | Governance Policy Compliance Audit | ISO 27001 |
| 16 | Audit-Reports-KPI **(Audit-Melaporkan-KPI)** | Association | MetadataAudit → KPIIndicator | 1..\* ↔ 0..\* | value:decimal; unit:string; period:dateRange | context MetadataAudit inv: self.kpis->notEmpty() | Audits produce indicators for governance reporting. | 2024 audit → 90% compliance KPI | COBIT 2019 |
| 17 | Risk-Impacts-Quality **(Risiko-BerdampakPada-Kualitas)** | Association | MetadataRisk → MetadataQuality | 1..\* ↔ 1..\* | impact:{H/M/L} | context MetadataRisk inv: self.affectedQuality->notEmpty() | Risks affect quality dimensions. | Integrity risk → consistency | ISO 27001 |
| 18 | Risk-MonitoredBy-Audit **(Risiko-DimonitorOleh-Audit)** | Association | MetadataRisk → MetadataAudit | 1..\* ↔ 1..\* | — | context MetadataRisk inv: self.monitoringAudits->notEmpty() | Risk controls are monitored by audit. | Risk mitigation audit | DAMA-DMBOK2 |
| 19 | ChangeRequest-Relates-Issue **(PermintaanPerubahan-Terkait-Isu)** | Association *(may be AssociationClass)* | MetadataChangeRequest → MetadataIssueLog | 1..\* ↔ 0..\* | relationType:{originatesFrom,duplicates,blocks} | context MetadataChangeRequest inv: self.issues->notEmpty() | Change requests are typically triggered by recorded issues. | Validation issue → CR #123 | DAMA-DMBOK2 |
| 20 | Lifecycle-Impacts-Change **(SiklusHidup-BerdampakPada-Perubahan)** | Association | LifecyclePhase → MetadataChangeRequest | 1..\* ↔ 0..\* | phase:{design,publication,deprecation,decommission} | context LifecyclePhase inv: self.changeRequests->notEmpty() | Lifecycle phases drive change needs. | Decommission → deletion CR | ISO/IEC 11179 |
| 21 | Lifecycle-TrackedBy-Audit **(SiklusHidup-DilacakOleh-Audit)** | Association | LifecyclePhase → MetadataAudit | 1..\* ↔ 0..\* | checkpoint:string | context LifecyclePhase inv: self.audits->notEmpty() | Each lifecycle phase has audit checkpoints. | Publication-phase audit | DAMA-DMBOK2 |
| 22 | KPI-Measures-Goal **(KPI-Mengukur-Tujuan)** | Association | KPIIndicator → MetadataGovernanceGoal | 1..\* ↔ 1..\* | target:decimal; threshold:decimal; direction:{↑,↓} | context KPIIndicator inv: self.goal->notEmpty() | KPIs measure attainment of governance goals. | Compliance KPI → quality goal | COBIT 2019 |
| 23 | Organization-Performs-Audit **(Organisasi-Melaksanakan-Audit)** | Association | MetadataOrganization → MetadataAudit | 1 ↔ 0..\* | unit:string | context MetadataOrganization inv: self.audits->notEmpty() | Audits are executed/led by the organization (actor). | Compliance Unit → 2024 Audit | ISO 27001; COBIT 2019 |
| 24 | Organization-Governs-Lifecycle **(Organisasi-Mengatur-SiklusHidup)** | Association | MetadataOrganization → LifecyclePhase | 1 ↔ 0..\* | policyRef:string | context MetadataOrganization inv: self.lifecyclePhases->notEmpty() | Organization governs execution of lifecycle phases. | Data Division → publication/decommission | ISO/IEC 11179; DAMA-DMBOK2 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Relation Name (EN)** | **Nama Relasi (IDN)** | **Tipe** | **Sumber→Target** | **Kardinalitas** | **Atribut Relasi (opsional)** | **OCL (ringkas)** | **Justifikasi Ontologis** | **Contoh Instansiasi** | **Rujukan** |
| 1 | Policy-Defines-Standard | Kebijakan-Mendefinisikan-Standar | Association | MetadataPolicy → MetadataStandard | 1..\* ↔ 1..\* | basis:string, effectiveDate:date | context MetadataPolicy inv: self.standards->size()>=1 | Normatif: kebijakan menurunkan standar teknis; bukan hierarki kelas. | “Kebijakan Metadata Nasional” → ISO/IEC 11179 | DAMA-DMBOK2; ISO/IEC 11179 |
| 2 | Policy-Defines-Rule | Kebijakan-Mendefinisikan-Aturan | Association | MetadataPolicy → MetadataRule | 1..\* ↔ 1..\* | rationale:string | context MetadataPolicy inv: self.rules->size()>=1 | Aturan operasional adalah turunan konkrit dari kebijakan. | Policy “Penamaan” → Rule “≤50 char” | DAMA-DMBOK2 |
| 3 | Policy-Ensures-Quality | Kebijakan-Memastikan-Kualitas | Association | MetadataPolicy → MetadataQuality | 1..\* ↔ 0..\* | target:decimal, operator:{≥,≤,=}, unit:string | context MetadataPolicy inv: self.qualityTargets->notEmpty() | Kebijakan menetapkan target kualitas (accuracy, completeness, dsb.). | Policy Kualitas → target completeness ≥95% | DAMA-DMBOK2 ch.13 |
| 4 | Goal-Guides-Policy | Tujuan-Mengarahkan-Kebijakan | Association | MetadataGovernanceGoal → MetadataPolicy | 1 ↔ 1..\* |  | context MetadataPolicy inv: self.goal->notEmpty() | Tujuan tata kelola mengarahkan pembentukan kebijakan. | Goal interoperabilitas → kebijakan standardisasi | ISO/IEC 38500 |
| 5 | Compliance-Constrains-Policy | SumberKepatuhan-Membatasi-Kebijakan | Association | ComplianceSource → MetadataPolicy | 1..\* ↔ 1..\* | clause:string | context MetadataPolicy inv: self.complianceSources->size()>=1 | Regulasi/standar eksternal membatasi/menjadi acuan kebijakan. | GDPR → Kebijakan Retensi | GDPR; ISO 27001 |
| 6 | Compliance-Constrains-Rule | SumberKepatuhan-Membatasi-Aturan | Association | ComplianceSource → MetadataRule | 1..\* ↔ 1..\* | clause:string | context MetadataRule inv: self.complianceSources->size()>=1 | Aturan spesifik sering bersumber dari regulasi. | GDPR → Rule penyimpanan data | GDPR |
| 7 | Rule-Requires-AccessControl | Aturan-Memerlukan-KontrolAkses | Association | MetadataRule → MetadataAccessControl | 1..\* ↔ 1..\* | controlType:string | context MetadataRule inv: self.accessControls->size()>=1 | Penerapan rule membutuhkan kontrol akses berdasar peran/kelas data. | Rule akses-atribut → ACL/ABAC | ISO/IEC 27001 |
| 8 | Rule-VerifiedBy-Audit | Aturan-DiverifikasiOleh-Audit | Association | MetadataRule → MetadataAudit | 1..\* ↔ 1..\* | auditType:string, sample:int | context MetadataRule inv: self.audits->size()>=1 | Kepatuhan rule diverifikasi melalui audit. | Audit 2024 memeriksa Rule penamaan | COBIT 2019; ISO 27001 |
| 9 | Organization-Assigns-Stewardship | Organisasi-Menetapkan-Stewardship | Association (dapat jadi AssociationClass) | MetadataOrganization → MetadataStewardship | 1 ↔ 1..\* | mandateDoc:string, appointmentDate:date, scope:string | context MetadataOrganization inv: self.stewardships->size()>=1 | Mandat stewardship merupakan keputusan organisasi (aktor). | BPS menetapkan Stewardship Katalog Metadata | DAMA-DMBOK2 |
| 10 | Stewardship-Has-Owner | Stewardship-Memiliki-Pemilik | Association | MetadataStewardship → MetadataOwner | 1..\* ↔ 1..1 | startDate:date | context MetadataStewardship inv: self.owner->size()=1 | Setiap stewardship memiliki pemilik tunggal yang akuntabel. | Stewardship A → Owner “Dir. Statistik” | Peraturan internal |
| 11 | Owner-Has-Role | Pemilik-Memiliki-Peran | Association | MetadataOwner → OwnerRole | 1 ↔ 1..\* | roleType:{DataOwner, Custodian, Steward} | context MetadataOwner inv: self.roles->notEmpty() | Seorang owner menjalankan satu atau lebih peran formal. | Owner X → Role Data Custodian | ISO/IEC 38500 |
| 12 | Owner-Oversees-Quality | Pemilik-Mengawasi-Kualitas | Association | MetadataOwner → MetadataQuality | 1 ↔ 1..\* |  | context MetadataOwner inv: self.qualityTargets->notEmpty() | Akuntabilitas kualitas ditetapkan kepada owner. | Owner menetapkan target completeness | DAMA-DMBOK2 |
| 13 | Owner-Authorizes-Change | Pemilik-Mengesahkan-Perubahan | Association | MetadataOwner → MetadataChangeRequest | 1 ↔ 0..\* | decision:{approve,reject}, decisionDate:date | context MetadataChangeRequest inv: self.authorizedBy->size()<=1 | Perubahan definisi harus disahkan owner. | Owner menyetujui perubahan definisi | COBIT 2019 |
| 14 | Owner-Resolves-Issue | Pemilik-Menyelesaikan-Isu | Association | MetadataOwner → MetadataIssueLog | 1 ↔ 0..\* | resolutionDate:date, severity:{low,med,high} | context MetadataIssueLog inv: self.resolvedBy->size()<=1 | Isu terkait metadata ditangani oleh owner yang relevan. | Owner menyelesaikan isu validasi | DAMA-DMBOK2 |
| 15 | Audit-Validates-Policy | Audit-Memvalidasi-Kebijakan | Association | MetadataAudit → MetadataPolicy | 1..\* ↔ 1..\* | auditPeriod:dateRange | context MetadataAudit inv: self.policies->notEmpty() | Audit menilai kepatuhan kebijakan. | Audit Kepatuhan Kebijakan Metadata | ISO 27001 |
| 16 | Audit-Reports-KPI | Audit-Melaporkan-KPI | Association | MetadataAudit → KPIIndicator | 1..\* ↔ 0..\* | value:decimal, unit:string, period:dateRange | context MetadataAudit inv: self.kpis->notEmpty() | Audit menghasilkan indikator kinerja untuk pelaporan governance. | Audit 2024 → KPI kepatuhan 90% | COBIT 2019 |
| 17 | Risk-Impacts-Quality | Risiko-BerdampakPada-Kualitas | Association | MetadataRisk → MetadataQuality | 1..\* ↔ 1..\* | impact:{H/M/L} | context MetadataRisk inv: self.affectedQuality->notEmpty() | Risiko memengaruhi dimensi kualitas. | Risiko integritas → konsistensi metadata | ISO 27001 |
| 18 | Risk-MonitoredBy-Audit | Risiko-DimonitorOleh-Audit | Association | MetadataRisk → MetadataAudit | 1..\* ↔ 1..\* |  | context MetadataRisk inv: self.monitoringAudits->notEmpty() | Pengendalian risiko dimonitor melalui audit. | Audit mitigasi risiko metadata | DAMA-DMBOK2 |
| 19 | ChangeRequest-Relates-Issue | PermintaanPerubahan-TerkaitDengan-Isu | Association (dapat jadi AssociationClass) | MetadataChangeRequest → MetadataIssueLog | 1..\* ↔ 0..\* | relationType:{originatesFrom,duplicates,blocks} | context MetadataChangeRequest inv: self.issues->notEmpty() | Permintaan perubahan biasanya muncul dari isu-isu terdokumentasi. | Isu validasi → CR #123 | DAMA-DMBOK2 |
| 20 | Lifecycle-Impacts-Change | SiklusHidup-BerdampakPada-Perubahan | Association | LifecyclePhase → MetadataChangeRequest | 1..\* ↔ 0..\* | phase:{design,publication,deprecation,decommission} | context LifecyclePhase inv: self.changeRequests->notEmpty() | Fase siklus hidup menentukan kebutuhan perubahan. | Dekomisioning → CR penghapusan metadata | ISO/IEC 11179 |
| 21 | Lifecycle-TrackedBy-Audit | SiklusHidup-DilacakOleh-Audit | Association | LifecyclePhase → MetadataAudit | 1..\* ↔ 0..\* | checkpoint:string | context LifecyclePhase inv: self.audits->notEmpty() | Setiap fase lifecycle memiliki checkpoint audit. | Audit fase publikasi | DAMA-DMBOK2 |
| 22 | KPI-Measures-Goal | KPI-Mengukur-Tujuan | Association | KPIIndicator → MetadataGovernanceGoal | 1..\* ↔ 1..\* | target:decimal, threshold:decimal, direction:{↑,↓} | context KPIIndicator inv: self.goal->notEmpty() | KPI mengukur ketercapaian goal tata kelola. | KPI kepatuhan → Goal kualitas | COBIT 2019 |
| 23 | Organization-Performs-Audit | Organisasi-Melaksanakan-Audit | Association | MetadataOrganization → MetadataAudit | 1 ↔ 0..\* | unit:string | context MetadataOrganization inv: self.audits->notEmpty() | Audit dilaksanakan/dikoordinasi oleh organisasi (aktor). | Unit Kepatuhan → Audit 2024 | ISO 27001; COBIT 2019 |
| 24 | Organization-Governs-Lifecycle | Organisasi-Mengatur-SiklusHidup | Association | MetadataOrganization → LifecyclePhase | 1 ↔ 0..\* | policyRef:string | context MetadataOrganization inv: self.lifecyclePhases->notEmpty() | Organisasi mengatur pelaksanaan fase siklus hidup metadata. | Divisi Data → fase publikasi/dekomisioning | ISO/IEC 11179; DAMA-DMBOK2 |

| **#** | **Nama Relasi** | **Tipe** | **Sumber→Target** | **Kardinalitas** | **Atribut Relasi (ops.)** | **OCL (ringkas)** | **Justifikasi Ontologis** | **Contoh Instansiasi** | **Rujukan** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Policy-Defines-Standard | Association | MetadataPolicy → MetadataStandard | 1..\* ↔ 1..\* | basis:string, effectiveDate:date | context MetadataPolicy inv: self.standards->size()>=1 | Kebijakan menurunkan standar teknis; bukan relasi pewarisan. | “Kebijakan Metadata Nasional” → ISO/IEC 11179 | DAMA-DMBOK2; ISO/IEC 11179 |
| 2 | Policy-Defines-Rule | Association | MetadataPolicy → MetadataRule | 1..\* ↔ 1..\* | rationale:string | context MetadataPolicy inv: self.rules->size()>=1 | Aturan operasional adalah turunan konkrit dari kebijakan. | Policy “Penamaan” → Rule “≤50 char” | DAMA-DMBOK2 |
| 3 | Policy-Ensures-Quality | Association | MetadataPolicy → MetadataQuality | 1..\* ↔ 0..\* | target:decimal, operator:{≥,≤,=}, unit:string | context MetadataPolicy inv: self.qualityTargets->notEmpty() | Kebijakan menetapkan target kualitas (accuracy, completeness, dsb.). | Policy Kualitas → target completeness ≥95% | DAMA-DMBOK2 ch.13 |
| 4 | Goal-Guides-Policy | Association | MetadataGovernanceGoal → MetadataPolicy | 1 ↔ 1..\* | — | context MetadataPolicy inv: self.goal->notEmpty() | Tujuan governance mengarahkan kebijakan. | Goal interoperabilitas → kebijakan standardisasi | ISO/IEC 38500 |
| 5 | Compliance-Constrains-Policy | Association | ComplianceSource → MetadataPolicy | 1..\* ↔ 1..\* | clause:string | context MetadataPolicy inv: self.complianceSources->size()>=1 | Regulasi/standar eksternal membatasi/mengarahkan kebijakan. | GDPR → Kebijakan Retensi | GDPR; ISO 27001 |
| 6 | Compliance-Constrains-Rule | Association | ComplianceSource → MetadataRule | 1..\* ↔ 1..\* | clause:string | context MetadataRule inv: self.complianceSources->size()>=1 | Aturan spesifik bersumber pada regulasi. | GDPR → Rule penyimpanan data | GDPR |
| 7 | Rule-Requires-AccessControl | Association | MetadataRule → MetadataAccessControl | 1..\* ↔ 1..\* | controlType:string | context MetadataRule inv: self.accessControls->size()>=1 | Penerapan rule memerlukan kontrol akses (role/kelas data). | Rule akses-atribut → ACL/ABAC | ISO/IEC 27001 |
| 8 | Rule-VerifiedBy-Audit | Association | MetadataRule → MetadataAudit | 1..\* ↔ 1..\* | auditType:string, sample:int | context MetadataRule inv: self.audits->size()>=1 | Kepatuhan rule diverifikasi melalui audit. | Audit 2024 memeriksa Rule penamaan | COBIT 2019; ISO 27001 |
| 9 | Organization-Assigns-Stewardship | Association *(dapat menjadi AssociationClass)* | MetadataOrganization → MetadataStewardship | 1 ↔ 1..\* | mandateDoc:string, appointmentDate:date, scope:string | context MetadataOrganization inv: self.stewardships->size()>=1 | Mandat stewardship adalah keputusan organisasi (aktor). | BPS menetapkan Stewardship Katalog Metadata | DAMA-DMBOK2 |
| 10 | Stewardship-Has-Owner | Association | MetadataStewardship → MetadataOwner | 1..\* ↔ 1..1 | startDate:date | context MetadataStewardship inv: self.owner->size()=1 | Stewardship memiliki satu pemilik akuntabel. | Stewardship A → Owner “Dir. Statistik” | Peraturan internal |
| 11 | Owner-Has-Role | Association | MetadataOwner → OwnerRole | 1 ↔ 1..\* | roleType:{DataOwner,Custodian,Steward} | context MetadataOwner inv: self.roles->notEmpty() | Owner menjalankan ≥1 peran formal. | Owner X → Role Data Custodian | ISO/IEC 38500 |
| 12 | Owner-Oversees-Quality | Association | MetadataOwner → MetadataQuality | 1 ↔ 1..\* | — | context MetadataOwner inv: self.qualityTargets->notEmpty() | Akuntabilitas kualitas ditetapkan ke owner. | Owner menetapkan target completeness | DAMA-DMBOK2 |
| 13 | Owner-Authorizes-Change | Association | MetadataOwner → MetadataChangeRequest | 1 ↔ 0..\* | decision:{approve,reject}, decisionDate:date | context MetadataChangeRequest inv: self.authorizedBy->size()<=1 | Perubahan definisi harus disahkan owner. | Owner menyetujui perubahan definisi | COBIT 2019 |
| 14 | Owner-Resolves-Issue | Association | MetadataOwner → MetadataIssueLog | 1 ↔ 0..\* | resolutionDate:date, severity:{L,M,H} | context MetadataIssueLog inv: self.resolvedBy->size()<=1 | Isu metadata ditangani owner terkait. | Owner menyelesaikan isu validasi | DAMA-DMBOK2 |
| 15 | Audit-Validates-Policy | Association | MetadataAudit → MetadataPolicy | 1..\* ↔ 1..\* | auditPeriod:dateRange | context MetadataAudit inv: self.policies->notEmpty() | Audit menilai kepatuhan kebijakan. | Audit Kepatuhan Kebijakan Metadata | ISO 27001 |
| 16 | Audit-Reports-KPI | Association | MetadataAudit → KPIIndicator | 1..\* ↔ 0..\* | value:decimal, unit:string, period:dateRange | context MetadataAudit inv: self.kpis->notEmpty() | Audit menghasilkan KPI untuk pelaporan governance. | Audit 2024 → KPI kepatuhan 90% | COBIT 2019 |
| 17 | Risk-Impacts-Quality | Association | MetadataRisk → MetadataQuality | 1..\* ↔ 1..\* | impact:{H/M/L} | context MetadataRisk inv: self.affectedQuality->notEmpty() | Risiko memengaruhi dimensi kualitas. | Risiko integritas → konsistensi | ISO 27001 |
| 18 | Risk-MonitoredBy-Audit | Association | MetadataRisk → MetadataAudit | 1..\* ↔ 1..\* | — | context MetadataRisk inv: self.monitoringAudits->notEmpty() | Pengendalian risiko dimonitor audit. | Audit mitigasi risiko metadata | DAMA-DMBOK2 |
| 19 | ChangeRequest-Relates-Issue | Association *(dpt jd AssociationClass)* | MetadataChangeRequest → MetadataIssueLog | 1..\* ↔ 0..\* | relationType:{originatesFrom,duplicates,blocks} | context MetadataChangeRequest inv: self.issues->notEmpty() | CR umumnya muncul dari isu terdokumentasi. | Isu validasi → CR #123 | DAMA-DMBOK2 |
| 20 | Lifecycle-Impacts-Change | Association | LifecyclePhase → MetadataChangeRequest | 1..\* ↔ 0..\* | phase:{design,publication,deprecation,decommission} | context LifecyclePhase inv: self.changeRequests->notEmpty() | Fase lifecycle memicu perubahan. | Dekomisioning → CR hapus metadata | ISO/IEC 11179 |
| 21 | Lifecycle-TrackedBy-Audit | Association | LifecyclePhase → MetadataAudit | 1..\* ↔ 0..\* | checkpoint:string | context LifecyclePhase inv: self.audits->notEmpty() | Tiap fase memiliki checkpoint audit. | Audit fase publikasi | DAMA-DMBOK2 |
| 22 | KPI-Measures-Goal | Association | KPIIndicator → MetadataGovernanceGoal | 1..\* ↔ 1..\* | target:decimal, threshold:decimal, direction:{↑,↓} | context KPIIndicator inv: self.goal->notEmpty() | KPI mengukur ketercapaian goal. | KPI kepatuhan → Goal kualitas | COBIT 2019 |
| 23 | Organization-Performs-Audit | Association | MetadataOrganization → MetadataAudit | 1 ↔ 0..\* | unit:string | context MetadataOrganization inv: self.audits->notEmpty() | Audit dilaksanakan/dikoordinasi organisasi (aktor). | Unit Kepatuhan → Audit 2024 | ISO 27001; COBIT 2019 |
| 24 | Organization-Governs-Lifecycle | Association | MetadataOrganization → LifecyclePhase | 1 ↔ 0..\* | policyRef:string | context MetadataOrganization inv: self.lifecyclePhases->notEmpty() | Organisasi mengatur pelaksanaan fase siklus hidup. | Divisi Data → fase publikasi/dekomisioning | ISO/IEC 11179; DAMA-DMBOK2 |

**Catatan:** Pada relasi #9 dan #19 saya beri catatan “dapat menjadi AssociationClass” karena relasi tersebut secara praktik sering memerlukan atribut relasi (mis. *mandateDoc*, *relationType*). Jika Anda ingin, saya bisa buatkan versi **AssociationClass** di file .drawio.

Metamodel\_Governance\_Clusters\_WithRelations.drawio

Isi:

* 4 klaster + **17 elemen** persis seperti versi clean.
* **22 relasi lintas-klaster** sesuai daftar lengkap:
  1. Policy→Standard, 2) Policy→Rule, 3) Policy→Quality,
  2. Goal→Policy, 5) Compliance→Policy, 6) Compliance→Rule,
  3. Rule→Access Control, 8) Rule→Audit,
  4. Organization→Stewardship, 10) Stewardship→Owner, 11) Owner→Role,
  5. Owner→Quality, 13) Owner→Change Request, 14) Owner→Issue Log,
  6. Audit→Policy, 16) Audit→KPI,
  7. Risk→Quality, 18) Risk→Audit,
  8. Change Request→Issue Log, 20) Lifecycle→Change Request, 21) Lifecycle→Audit,
  9. KPI→Goal.

==

**1. Mengapa Audit tidak memiliki relasi langsung dengan klaster Organizational Accountability?**

* **Struktur dalam katalog:** Audit ditempatkan sebagai elemen *Assurance & Risk Control*. Relasinya ke *Policy* (validasi kebijakan), *Rule* (validasi aturan), *Risk* (pemantauan risiko), *Lifecycle* (audit fase siklus hidup), dan *KPI* (hasil audit).
* **Alasan ontologis:** Audit dalam tata kelola metadata berfungsi sebagai **mekanisme pengawasan eksternal** yang memvalidasi *proses* dan *artefak kebijakan/aturan*, bukan sebagai mekanisme langsung untuk memeriksa aktor organisasi (steward/owner). Audit menilai apakah *proses* yang dijalankan sesuai kebijakan dan standar, sementara **akuntabilitas organisasi sudah diwujudkan melalui relasi Organization–Stewardship–Owner–Role**.
* **Bandingkan:** Dalam COBIT 2019, audit diposisikan sebagai mekanisme *evaluate–direct–monitor* yang fokus ke *policies, practices, risks*, bukan ke peran manusia secara langsung.
* **Implikasi:** Relasi audit ke organisasi bisa dianggap **implisit** (audit menilai apakah steward/owner menjalankan tugas sesuai peran), tapi secara konseptual hal ini sudah tercermin lewat audit terhadap kebijakan/aturan yang menjadi mandat organisasi.

**2. Mengapa tidak ada relasi langsung antara Organization dan Lifecycle?**

* **Struktur dalam katalog:** Lifecycle berada di klaster *Governance Execution* dan berelasi dengan *ChangeRequest* serta *Audit*. Organization tidak terhubung langsung ke Lifecycle.
* **Alasan ontologis:**
  + Lifecycle adalah **fase keberadaan metadata** (misalnya registrasi, publikasi, dekomisioning).
  + Organization adalah **struktur akuntabilitas peran**.
  + Secara ontologis, lifecycle adalah **proses entitas metadata**, sedangkan organization adalah **aktor**.
* **Alasan metodologis:** Agar relasi dalam metamodel tetap konsisten, kita pisahkan antara *aktor* (organization–owner–role) dan *proses metadata* (lifecycle–change–issue).
* **Namun, hubungan praktisnya ada:** misalnya, *organization menetapkan siapa yang bertanggung jawab pada setiap fase lifecycle*. Itu berarti organisasi berhubungan dengan lifecycle melalui **mediator: owner atau steward**.
* **Implikasi:** Jika kita tambahkan relasi langsung *Organization → Lifecycle*, akan menimbulkan redundansi karena sudah terwakili melalui *Owner → ChangeRequest/Issue* dan *Lifecycle → Audit*.

**🔹 Kesimpulan Diskusi**

* Tidak adanya relasi langsung Audit–Organization dan Organization–Lifecycle adalah **pilihan desain konseptual untuk menjaga kejernihan metamodel**.
* Audit → Organization bisa dianggap **implisit** melalui audit terhadap kebijakan/aturan yang dijalankan oleh aktor.
* Organization → Lifecycle bisa dianggap **implisit** melalui steward/owner yang menjalankan aktivitas pada setiap fase lifecycle.
* Jika pada saat validasi pakar muncul pertanyaan yang sama, penjelasan ini bisa Anda gunakan:

*“Metamodel ini memisahkan antara dimensi akuntabilitas (aktor) dan dimensi proses metadata. Hubungan audit ke aktor, atau organisasi ke lifecycle, dipandang implisit melalui kebijakan, aturan, dan peran yang sudah dimodelkan. Hal ini menghindari redundansi relasi dan menjaga konsistensi struktur ontologis.”*

**🔹 Menanggapi Argumen “Organization sebagai Aktor Audit & Lifecycle”**

**1. Posisi Organization dalam Metamodel**

* Dalam metamodel metadata governance, **Organization, Stewardship, Owner, dan Role** membentuk klaster *Organizational Accountability*.
* Ontologinya: Organization adalah **aktor struktural** yang mendefinisikan *peran dan akuntabilitas*, bukan pelaksana teknis langsung dari proses (audit, lifecycle, change).
* Dengan kata lain, Organization *menugaskan*, *menunjuk*, dan *mengatur*, tetapi eksekusi audit atau pengelolaan siklus hidup metadata dilakukan oleh mekanisme/proses (Audit, LifecyclePhase) yang berada di klaster lain.

**2. Mengapa Audit tidak langsung dikaitkan dengan Organization**

* Audit memang *diinisiasi oleh organisasi* (misalnya BPS, Kemenkeu).
* Namun, secara ontologis, audit dalam governance dipandang sebagai **mekanisme pengawasan independen**, bukan tindakan langsung aktor.
* Jika relasi Audit → Organization dimodelkan, maka fungsi audit akan “terserap” sebagai aktivitas organisasi tertentu, padahal prinsip audit dalam COBIT 2019 dan ISO/IEC 27001 adalah **independen dari struktur organisasi** untuk menjaga obyektivitas.
* Oleh sebab itu, metamodel memisahkan:
  + Organization mengatur **siapa steward/owner**.
  + Audit menguji **kebijakan, aturan, risiko, dan kualitas**.
  + Keterhubungan Organization–Audit **implisit**: steward/owner lah yang menjadi obyek audit melalui kepatuhan mereka pada kebijakan/aturan.

**3. Mengapa Lifecycle tidak langsung dikaitkan dengan Organization**

* Lifecycle menggambarkan **fase keberadaan metadata** (registrasi, penggunaan, dekomisioning).
* Governance lifecycle memang *dikelola oleh organisasi*, tetapi peran itu sudah diwakili oleh **Owner/Steward** yang bertanggung jawab di tiap fase.
* Jika dibuat relasi langsung Organization–Lifecycle, hasilnya akan redundan, karena sudah ada jalur:
  + Organization → Stewardship → Owner → (mengelola) ChangeRequest/Issue,
  + Lifecycle → Change/Audit (yang memverifikasi fase).
* Dengan kata lain, Organization sudah *mewakilkan* governance lifecycle kepada pemilik/steward.

**4. Jawaban ke penguji (narasi singkat siap pakai)**

*“Dalam metamodel ini Organization diposisikan sebagai entitas akuntabilitas yang menetapkan peran (Steward, Owner, Role). Eksekusi proses audit maupun lifecycle tidak ditempelkan langsung pada Organization untuk menjaga konsistensi ontologis: audit adalah mekanisme independen, sementara lifecycle adalah sifat dari metadata itu sendiri. Hubungan Organization ke audit dan lifecycle dipandang implisit, dimediasi oleh steward/owner yang telah ditetapkan. Jika kita hubungkan langsung, akan muncul redundansi relasi dan bias pada prinsip independensi audit.”*

**🔹 Apakah Klaster Organizational Accountability Harus Terhubung ke Audit dan Lifecycle?**

**1. Hubungan dengan Audit**

* **Argumen penguji**: audit pasti dilakukan oleh organisasi, jadi klaster Organization harus punya relasi eksplisit ke Audit.
* **Jawaban konseptual**:
  + Benar bahwa audit *dijalankan dalam konteks organisasi*. Namun, di level metamodel, audit diposisikan sebagai **mekanisme kontrol normatif**, bukan aktivitas yang melekat pada struktur organisasi tertentu.
  + Relasi sudah ada secara **implisit**: steward/owner (aktor organisasi) yang bertanggung jawab atas kualitas metadata **menjadi obyek audit**. Audit → Policy/Rule → dijalankan oleh steward/owner.
  + Menjadikan relasi langsung *Organization → Audit* berisiko menabrak prinsip **independensi audit** (COBIT 2019, ISO/IEC 27001), karena audit seharusnya tidak “dimiliki” oleh organisasi yang diaudit.

**2. Hubungan dengan Lifecycle**

* **Argumen penguji**: siklus hidup metadata pasti dikelola organisasi, maka klaster Organization seharusnya terhubung dengan LifecyclePhase.
* **Jawaban konseptual**:
  + Lifecycle adalah sifat dari metadata sebagai entitas: registrasi, publikasi, pemeliharaan, dekomisioning.
  + Organization memang **bertanggung jawab** mengelola lifecycle, tapi tanggung jawab itu sudah diwujudkan melalui *Owner dan Stewardship*.
  + Relasi eksplisit *Owner → ChangeRequest/Issue* dan *Lifecycle → Change/Audit* sudah cukup mewakili. Kalau Organization → Lifecycle ditambahkan, akan terjadi **redundansi** (dua jalur untuk hal yang sama).

**3. Prinsip Desain Metamodel**

* Klaster Organizational Accountability di sini memang diposisikan sebagai **aktor dan akuntabilitas**.
* Klaster Assurance & Risk Control dan Governance Execution adalah **mekanisme/proses**.
* Jadi, filosofi desainnya adalah **pemisahan ontologis**: aktor mengatur dan bertanggung jawab, tapi tidak langsung “menjadi” mekanisme audit atau lifecycle.

**🔹 Narasi Singkat untuk Penguji**

*“Memang benar bahwa audit dan lifecycle tidak bisa dilepaskan dari organisasi, karena aktor organisasi yang melakukannya. Namun, dalam metamodel ini kami memisahkan antara klaster aktor (Organizational Accountability) dengan klaster mekanisme (Assurance & Risk Control dan Governance Execution). Relasi aktor ke audit/lifecycle dipandang implisit, dimediasi oleh Steward dan Owner. Dengan cara ini, kami menjaga konsistensi ontologis: audit tetap independen, lifecycle tetap proses metadata, sementara organisasi tetap entitas akuntabilitas. Jika dibuat eksplisit, akan ada redundansi dan potensi bias konseptual.”*

Relasi nambah audit dengan organisasi

Metamodel\_Governance\_Clusters\_Expansive.drawio

Isi sbb:

**🔹 Perbandingan Versi Inti dan Ekspansif Relasi Metamodel Metadata Governance**

**1. Versi Inti (22 Relasi)**

* **Desain:** hanya memasukkan relasi yang benar-benar memiliki **landasan ontologis eksplisit** dalam literatur (DAMA-DMBOK2, ISO/IEC 11179, COBIT 2019, ISO 27001, ISO 38500, GDPR, Perpres 39/2019).
* **Filosofi:** memisahkan secara tegas antara **aktor/akuntabilitas** (klaster *Organizational Accountability*) dan **mekanisme/proses** (klaster *Assurance & Risk Control* serta *Governance Execution*).
* **Kelebihan:**
  + Struktur relasi sederhana dan tidak redundan.
  + Menjaga konsistensi ontologis, misalnya audit tetap independen dan lifecycle tetap dipandang sebagai sifat metadata, bukan milik organisasi.
* **Keterbatasan:**
  + Bisa menimbulkan pertanyaan dari pakar: “Mengapa organisasi tidak tampak seolah-olah melakukan audit atau mengelola lifecycle?”

**2. Versi Ekspansif (24 Relasi)**

* **Desain:** menambahkan dua relasi eksplisit:
  1. **Organization → Audit** (organisasi melaksanakan audit)
  2. **Organization → Lifecycle** (organisasi mengelola siklus hidup metadata)
* **Filosofi:** menekankan peran organisasi sebagai **aktor nyata** yang menjalankan mekanisme audit dan lifecycle.
* **Kelebihan:**
  1. Lebih mudah dipahami oleh praktisi, karena langsung menunjukkan organisasi sebagai pelaku audit dan pengelola siklus hidup.
  2. Mengurangi risiko pertanyaan penguji tentang “keterputusan” antara organisasi dan mekanisme.
* **Keterbatasan:**
  1. Berpotensi **redundan** (karena relasi ini sudah terwakili secara implisit melalui steward/owner dan audit terhadap kebijakan/aturan).
  2. Bisa menimbulkan perdebatan tentang **independensi audit**, karena audit semestinya bukan “milik” organisasi yang diaudit.

**3. Sintesis dan Posisi Penelitian**

* **Versi inti** digunakan sebagai **representasi konseptual resmi** dalam disertasi, karena menjaga konsistensi ontologis dan mengikuti prinsip desain metamodel konseptual.
* **Versi ekspansif** dapat ditampilkan sebagai **lampiran atau alternatif pandangan**, yang berguna saat menghadapi pertanyaan penguji atau diskusi pakar.
* Dengan cara ini, penelitian menunjukkan **fleksibilitas**: tetap konsisten secara teoretis, namun mampu menyesuaikan dengan kebutuhan praktis implementasi.

**4. Narasi Singkat untuk Sidang**

*“Dalam desain inti, metamodel ini memisahkan peran organisasi sebagai entitas akuntabilitas dari mekanisme audit dan lifecycle untuk menjaga konsistensi ontologis. Namun, kami juga menyiapkan versi ekspansif yang menambahkan relasi Organization–Audit dan Organization–Lifecycle. Versi ekspansif ini mempermudah pemahaman praktis, meski secara teoretis relasi tersebut sudah implisit. Dengan demikian, penelitian ini mengakomodasi kedua perspektif: kesahihan konseptual sekaligus keterbacaan praktis.”*