🧩 1. Purpose

Tujuan pengembangan ini adalah meningkatkan sistem pengelolaan **program On The Job Training (OJT)** agar dapat dikelola secara terintegrasi di Odoo 18 — mulai dari **penerimaan peserta, pengelolaan kegiatan dan agenda, pengumpulan tugas, absensi, hingga penerbitan e-certificate** yang dapat diakses publik.

🏗️ 2. Base Modules & Integration

Base Module	Pemanfaatan	Integrasi
Recruitment	Pendaftaran dan seleksi kandidat OJT	Linked dengan Employee dan OJT Batch
Event	Menangani jadwal kegiatan OJT, termasuk sesi kelas, mentoring, dan evaluasi	Terhubung dengan Attendance dan Assignment
Online Training (eLearning)	Modul pembelajaran, kuis, dan upload tugas	Sinkron dengan Event dan Participant Progress
Website	Publikasi lowongan OJT dan verifikasi e-certificate	Terhubung dengan portal publik
Survey/Assessment	Evaluasi hasil belajar dan kepuasan peserta	Terhubung dengan eCertificate scoring

---

⚙️ 3. Core Functional Requirements

3.1 Recruitment & Batch Management

Tujuan: Mengelola proses pendaftaran dan seleksi peserta OJT.

- R1.1: Sistem dapat menampilkan daftar **program OJT aktif** (periode, tema, divisi, kuota).
- R1.2: Kandidat dapat **mendaftar online** melalui portal Recruitment dengan melampirkan CV, form motivasi, dan pilihan batch.
- R1.3: Admin dapat melakukan **screening dan shortlisting** kandidat.

- R1.4: Status pendaftaran (Pending, Shortlisted, Accepted, Rejected) ditampilkan otomatis pada portal peserta.
- R1.5: Peserta diterima otomatis menjadi **OJT Participant** dan terhubung ke batch program.

3.2 Training & Event Scheduling

Tujuan: Mengatur jadwal kegiatan, mentor, dan agenda sesi.

- R2.1: Admin/Trainer dapat membuat **Event OJT** berdasarkan batch (topik, tanggal, instruktur, mode: online/offline).
- R2.2: Event dapat memiliki **sub-session/agenda** (misalnya: Orientation, Daily Standup, Review, Final Presentation).
- R2.3: Integrasi dengan **Online Meeting Link (MS Teams/Zoom/Google Meet)** otomatis.
- R2.4: Peserta dapat melihat agenda dan status kehadiran melalui portal.
- R2.5: Event otomatis mencatat daftar hadir (link attendance).

3.3 Attendance & Presence Tracking

Tujuan: Memonitor kehadiran peserta setiap sesi.

- R3.1: Sistem menyediakan **Attendance form** per event.
- R3.2: Absensi dapat dilakukan melalui:
  - QR Code (offline)
  - Auto log saat join meeting (online)
- R3.3: Sistem menghitung **persentase kehadiran peserta**.
- R3.4: Kehadiran otomatis memengaruhi eligibility eCertificate.

3.4 Task & Assignment Management

Tujuan: Mengelola tugas peserta per topik dan sesi.

- R4.1: Admin/Trainer dapat membuat **Assignment** per event/topik (upload file, deadline, skor).
- R4.2: Peserta dapat **mengunggah tugas** langsung melalui portal.
- R4.3: Trainer dapat memberikan **feedback dan skor** pada setiap tugas.
- R4.4: Status tugas peserta dapat dilihat di dashboard masing-masing.
- R4.5: Tugas dan skor berkontribusi pada total penilaian akhir (Final Score).

3.5 Participant Dashboard

Tujuan: Menyediakan tampilan progress peserta.

- R5.1: Menampilkan ringkasan:
  - Profil peserta
  - Batch dan jadwal
  - Attendance rate
  - Task completion rate
  - Score average

- R5.2: Peserta dapat mengunduh jadwal atau sertifikat (jika lulus).
- R5.3: Tersedia progress bar “OJT Completion (%)”.

3.6 eCertificate Generation

Tujuan: Mengeluarkan sertifikat digital otomatis.

- R6.1: eCertificate dihasilkan otomatis jika:
  - Kehadiran ≥ 80%
  - Nilai akhir ≥ ambang batas (contoh: 70)

- R6.2: eCertificate mencantumkan:
  - Nama peserta
  - Periode dan topik OJT
  - Nilai dan kategori kelulusan
  - QR Code verifikasi publik

- R6.3: Verifikasi eCertificate dapat dilakukan publik melalui halaman **/certificate/verify?code=XXXX**.
- R6.4: Template sertifikat dapat disesuaikan (brand Solvera, tanda tangan digital, watermark, dll).

3.7 Mentor & Trainer Management

Tujuan: Mengelola instruktur atau pembimbing.

- R7.1: Admin dapat mendaftarkan mentor internal atau eksternal.
- R7.2: Mentor dapat mengakses dashboard peserta batch-nya.

- R7.3: Mentor dapat memberi skor dan komentar terhadap tugas dan kehadiran.

3.8 Reporting & Analytics

Tujuan: Memberikan insight bagi admin dan manajemen Solvera.

- R8.1: Laporan kehadiran per batch dan per peserta.
- R8.2: Laporan nilai rata-rata dan penyebaran skor.
- R8.3: Export ke Excel/PDF.
- R8.4: Dashboard analitik OJT:
    - Jumlah batch aktif
    - Jumlah peserta per batch
    - Nilai rata-rata per topik
    - Tingkat kelulusan

3.9 Public Portal & Verification

Tujuan: Memberi akses publik untuk validasi sertifikat dan informasi program.

- R9.1: Halaman publik menampilkan:
    - Daftar batch OJT aktif
    - Detail program (tujuan, jadwal, mentor)
- R9.2: Fitur verifikasi eCertificate dengan memasukkan kode unik.
- R9.3: Hasil verifikasi menampilkan:
    - Nama peserta
    - Batch
    - Status (Valid/Expired/Invalid)

🧠 4. Non-Functional Requirements

Kategori	Deskripsi
Security	Akses data peserta dan nilai dilindungi role-based access control.
Performance	Dashboard harus mampu menampilkan progress hingga 500 peserta per batch.

Scalability	Sistem dapat menampung 50 batch aktif dan 2.000 peserta aktif.
Usability	Portal peserta responsif (desktop dan mobile).
Integration	API integration untuk e-certificate verification dan online meeting link.
Branding	Menggunakan warna dan logo korporat Solvera Indonesia.

📦 5. Expected Deliverables

1. **Custom Module:** `solvera_ojt_core`
2. **Extended Models:**
   - `ojt.batch`
   - `ojt.participant`
   - `ojt.assignment`
   - `ojt.attendance`
   - `ojt.certificate`
3. **Enhanced Views:**
   - OJT Batch Form & Kanban
   - Event Agenda View
   - Participant Dashboard (Portal)
   - Certificate Verification Page
4. **Reports:**
   - Attendance Summary
   - Score & Certificate Report

1) Ringkasan Arsitektur & Dependensi

Nama modul: `solvera_ojt_core`
Tujuan: Orkestrasi program OJT end-to-end (Recruitment → Batch → Event/Agenda → Assignment → Attendance → Scoring → e-Certificate publik).

Depends on:

- `hr_recruitment` (kandidat, lowongan)
- `event` (jadwal/sesi)
- `website_slides` (eLearning / course & content)
- `survey` (kuis/evaluasi)
- `portal` (akses peserta)
- `website` (halaman publik & verifikasi sertifikat)
- `mail` (chatter, aktivitas, template email)
- *(opsional)* `calendar`, `rating`, `utm` jika dibutuhkan

Model Odoo Terkait:

- `hr.applicant`, `hr.job`
- `event.event`, `event.registration`
- `slide.channel`, `slide.slide`
- `survey.survey`, `survey.user_input`
- `res.partner`, `res.users`

---

2) ERD (Gambaran Relasi)

```
hr.job (Lowongan) 1---* ojt.batch *---* res.partner (Mentor via M2M)
                           |
                           | 1---* ojt.participant *---1 res.partner (Peserta)
                           |              |
                           |              | *---* event.event (via ojt.event.link)
                           |              |
                           |              *---* slide.channel (course enrollment)
                           |
                           | 1---* ojt.assignment 1---* ojt.assignment.submit
                           |
                           | 1---* ojt.attendance (per event/session per participant)
                           |
                           | 1---* ojt.certificate (per participant per batch)
                           |
                           *---* survey.survey (evaluasi batch & final test) [opsional]
```

Keterangan tambahan:

- `event.event` merepresentasikan sesi/agenda OJT. Registrasi event menggunakan `event.registration`.
- Link banyak-ke-banyak antara batch dan event disimpan di `ojt.event.link` agar batch dapat mewarisi/menautkan beberapa event.
- `ojt.participant` mengikat peserta (`res.partner`) ke batch serta status progresnya.

---

3) Definisi Model & Field Detail

Notasi field:
`name: type [attrs] # help/notes`

3.1 `ojt.batch`

Deskripsi: Master program/batch OJT.

- `name: Char [required, index]` – Nama batch (mis. “OJT BA Oct 2025”)
- `code: Char [required, unique]` – Kode batch (sequence: `OJTBATCH/%(year)s/%(seq)05d`)
- `job_id: Many2one(hr.job)` – Lowongan/posisi terkait (untuk rekrutmen)
- `description: Html` – Deskripsi program
- `department_id: Many2one(hr.department)` – Divisi penyelenggara
- `mentor_ids: Many2many(res.partner, rel='ojt_batch_mentor_rel')` – Mentor/instruktur
- `start_date: Date [required]`
- `end_date: Date [required]`
- `mode: Selection([('online','Online'),('offline','Offline'),('hybrid','Hybrid')]) [default='online']`
- `capacity: Integer` – Kuota peserta target
- `participant_ids: One2many(ojt.participant, 'batch_id')`
- `event_link_ids: One2many(ojt.event.link, 'batch_id')` – Mapping ke event/event agenda
- `course_ids: Many2many(slide.channel, rel='ojt_batch_channel_rel')` – Course eLearning terkait
- `survey_id: Many2one(survey.survey)` – Survey evaluasi batch (opsional)

- `state: Selection([('draft','Draft'),('recruit','Recruitment'),('ongoing','Ongoing'),('done','Done'),('cancel','Cancelled')]) [default='draft', track_visibility='onchange']`
- `certificate_rule_attendance: Float [default=80.0]` – % minimal kehadiran
- `certificate_rule_score: Float [default=70.0]` – Skor minimal kelulusan
- `progress_ratio: Float [compute=_compute_progress_ratio, store=True]` – Progress rata-rata batch
- `color: Integer` – Untuk kanban color
- `company_id: Many2one(res.company) [default=current_company, required]`
- `active: Boolean [default=True]`

Constraints/Business Rules:

- `start_date <= end_date`
- `state` berubah otomatis:
  - Recruit → ketika dibuka pendaftaran (link ke `hr.applicant`)
  - Ongoing → ketika tanggal mulai tercapai / event pertama dimulai
  - Done → ketika semua agenda selesai & scoring final done
- Penguncian pengeditan komponen tertentu saat `state in ('ongoing','done')`.

3.2 `ojt.event.link`

Deskripsi: Tabel penghubung batch OJT dengan event/sesi (agenda).

- `batch_id: Many2one(ojt.batch) [required, ondelete='cascade']`
- `event_id: Many2one(event.event) [required]`
- `is_mandatory: Boolean [default=True]` – Wajib hadir untuk sertifikat
- `weight: Float [default=1.0]` – Bobot sesi untuk progress
- `online_meeting_url: Char` – Override/shortcut meet link (fallback ke event)
- `notes: Text`

3.3 `ojt.participant`

Deskripsi: Entri peserta per batch.

- `name: Char [compute from partner_id + batch, store=True, index=True]`
- `batch_id: Many2one(ojt.batch) [required, ondelete='cascade']`
- `partner_id: Many2one(res.partner) [required]` – Data peserta (portal)
- `applicant_id: Many2one(hr.applicant)` – Riwayat dari rekrutmen
- `registration_id: Many2one(event.registration)` – Jika memakai registrasi global
- `course_ids: Many2many(slide.channel, rel='ojt_participant_channel_rel')` – Enrollment eLearning
- `attendance_count: Integer [compute, store=True]`
- `attendance_rate: Float [compute, store=True]` – % hadir dari total sesi wajib
- `assignment_submit_ids: One2many(ojt.assignment.submit, 'participant_id')`
- `score_avg: Float [compute, store=True]`
- `score_final: Float [compute or write, store=True]` – Formula: weighted assignments + kuis (survey) + penilaian mentor
- `mentor_score: Float` – Skor penilaian akhir mentor (opsional)
- `state: Selection([('draft','Draft'),('active','Active'),('completed','Completed'),('failed','Failed'),('left','Left')]) [default='active']`
- `certificate_id: One2many(ojt.certificate, 'participant_id')`
- `portal_token: Char [index]` – Token akses cepat portal progress
- `notes: Text`
- `company_id: Many2one(res.company) [required]`

Rules:

- `attendance_rate` & `score_final` menentukan eligibility sertifikat.
- Jika `left`, peserta dikecualikan dari perhitungan kelulusan.

3.4 `ojt.assignment`

Deskripsi: Master tugas/penugasan per batch atau per event/topik.

- `name: Char [required]`
- `batch_id: Many2one(ojt.batch) [required]`
- `event_link_id: Many2one(ojt.event.link)` – Jika tugas terkait sesi tertentu
- `description: Html`
- `type: Selection([('task','Task/Project'),('quiz','Quiz (Survey/Slides)'),('presentation','Presentation')]) [default='task']`
- `related_channel_id: Many2one(slide.channel)` – Jika tugas berbasis course
- `deadline: Datetime`
- `max_score: Float [default=100.0]`
- `weight: Float [default=1.0]` – Bobot untuk final score
- `attachment_required: Boolean [default=True]`
- `submit_ids: One2many(ojt.assignment.submit, 'assignment_id')`
- `state: Selection([('draft','Draft'),('open','Open'),('closed','Closed')]) [default='open']`
- `company_id: Many2one(res.company) [required]`

Rules:

- Tidak bisa `closed` sebelum seluruh submission terskoring atau lewat grace period.

3.5 `ojt.assignment.submit`

Deskripsi: Submission tugas per peserta.

- `assignment_id: Many2one(ojt.assignment) [required, ondelete='cascade']`
- `participant_id: Many2one(ojt.participant) [required, ondelete='cascade']`
- `submitted_on: Datetime`
- `attachment_ids: Many2many(ir.attachment, rel='ojt_assignment_submit_attachment_rel')`
- `url_link: Char` – Jika submission berupa URL (Git, Figma, video)
- `score: Float`
- `reviewer_id: Many2one(res.users)` – Penilai/mentor
- `feedback: Html`
- `late: Boolean [compute]` – Dibanding `deadline`

- `state: Selection([('draft','Draft'),('submitted','Submitted'),('scored','Scored')]) [default='submitted']`

Rules:

- Skor tidak boleh > `assignment.max_score`.
- `late=True` jika `submitted_on > deadline`.

3.6 `ojt.attendance`

Deskripsi: Absensi kehadiran peserta per sesi (event).

- `batch_id: Many2one(ojt.batch) [required]`
- `event_link_id: Many2one(ojt.event.link) [required]`
- `event_id: Many2one(event.event) [required]`
- `participant_id: Many2one(ojt.participant) [required]`
- `check_in: Datetime`
- `check_out: Datetime`
- `presence: Selection([('present','Present'),('late','Late'),('absent','Absent')]) [default='present']`
- `method: Selection([('qr','QR Scan'),('online','Online Join'),('manual','Manual')]) [default='online']`
- `duration_minutes: Float [compute]`
- `notes: Text`

Rules:

- Otomasi `presence` (mis: <15 menit setelah start → `late`; >X menit tanpa check-in → `absent`).
- Unik: `(participant_id, event_id)`.

3.7 `ojt.certificate`

Deskripsi: Sertifikat digital per peserta per batch.

- `name: Char [required]` – Judul sertifikat (ex: “OJT Business Analyst – Oct 2025”)
- `batch_id: Many2one(ojt.batch) [required, index]`
- `participant_id: Many2one(ojt.participant) [required, index]`
- `partner_id: Many2one(res.partner) [related=participant_id.partner_id, store=True]`
- `serial: Char [required, unique, index]` – Kode unik (sequence: `OJTCERT/%(year)s/%(seq)06d`)
- `issued_on: Date [default=fields.Date.today]`
- `attendance_rate: Float [store=True]`
- `final_score: Float [store=True]`
- `grade: Selection([('A','A'),('B','B'),('C','C')]) [compute or write]`
- `qr_token: Char [required, unique, index]` – Token verifikasi publik (uuid)
- `pdf_report: Binary` – Render PDF sertifikat
- `state: Selection([('draft','Draft'),('issued','Issued'),('revoked','Revoked')]) [default='draft']`
- `notes: Text`

Rules:

- Hanya bisa `issued` jika memenuhi `batch.certificate_rule_attendance` & `certificate_rule_score`.
- Endpoint verifikasi menggunakan `qr_token`/`serial`.

---

4) Workflow & Aturan Bisnis Utama

1. **Recruitment → Participant**
   - Dari `hr.applicant` berstatus “Accepted” dapat dibuat `ojt.participant` terikat `ojt.batch`.
   - Otomatis kirim **portal invitation** ke peserta (template email).
2. **Batch → Event/Agenda**
   - Koordinator menautkan event via `ojt.event.link` dan tandai sesi wajib.
   - Bila `event` punya `join_url`, ditarik ke `online_meeting_url` bila kosong.

3. **Attendance**
    - Online: webhook/cron membaca kehadiran (opsional integrasi Teams/Zoom/Meet) → buat/isi `ojt.attendance`.
    - Offline: QR code check-in (controller kecil) → isi `check_in` & `presence`.
    - Hitung `attendance_rate` per peserta.

4. **Assignment**
    - Buat tugas, tentukan `deadline`, `weight`, `max_score`.
    - Peserta upload (file/URL), mentor memberi skor & feedback.
    - `score_final` dihitung: Σ(weighted assignment) + `mentor_score` + (opsional) nilai kuis `survey.user_input`.

5. **Certificate**
    - Wizard `Generate Certificates` pada `ojt.batch` memfilter peserta yang lulus.
    - Generate `serial`, `qr_token`, render PDF (QWeb), state → `issued`.
    - Email ke peserta + tautan verifikasi publik.

---

5) Security (Groups, ACL, Record Rules)

Groups:

- `ojt_group_manager` – full access semua objek OJT
- `ojt_group_trainer` – kelola assignment, attendance, scoring
- `ojt_group_mentor` – baca peserta batch-nya, nilai tugas
- `ojt_group_coordinator` – kelola batch, link event, certificate generation
- `ojt_group_viewer` – read-only report OJT
- `portal` – peserta melihat portal progres sendiri

ACL (contoh ringkas):

- `ojt.batch`: manager/coordinator RW, trainer R, mentor R, viewer R
- `ojt.event.link`: manager/coordinator RW, trainer R
- `ojt.participant`: manager/coordinator RW, trainer R, mentor R (hanya batch-nya), portal: read self
- `ojt.assignment`: manager/trainer RW, mentor R
- `ojt.assignment.submit`: manager/trainer/mentor RW (nilai), portal: create/write own
- `ojt.attendance`: manager/trainer RW, mentor R (read), portal: read own
- `ojt.certificate`: manager/coordinator RW, portal: read own (published)

Record Rules contoh:

- Mentor hanya baca `ojt.participant` & `ojt.assignment.submit` untuk `batch_id` yang memuatnya di `mentor_ids`.
- Portal user hanya akses record `participant.partner_id == user.partner_id`.

---

6) Tampilan (Views) & Menu

Menu Root: *OJT Management*
Submenu:

- Batches (Kanban/List/Form) – warna, progress, state buttons (Recruit/Ongoing/Done)
- Agenda (smart button dari Batch → `ojt.event.link` tree)
- Participants (List/Form) – dashboard mini: attendance%, assignments, score
- Assignments (List/Form) – kanban by state; smart button submissions
- Submissions (List/Form) – quick scoring UX (kanban)
- Attendance (List/Graph/Pivot)
- Certificates (List/Form) – print/download; wizard bulk issue
- Reports (Dashboard)

Portal:

- *My OJT* – Progress bar, agenda, tombol Join Meeting, tugas saya, status nilai, attendance%
- *Assignments* – daftar tugas + upload
- *Certificates* – unduh sertifikat jika issued

Website (publik):

- Halaman *OJT Programs* (daftar batch aktif)
- Halaman *Certificate Verify* (form input `serial`/`qr_token`) + hasil verifikasi

---

7) Laporan & Dashboard

- **Attendance Summary** (Pivot/Graph/Excel export): per batch/per event/per peserta.
- **Assignment Score Distribution** (bar chart).
- **Completion & Pass Rate** (kartu KPI).
- **Certificate Issued List** dengan filter batch/grade/periode.
- QWeb PDF: **Certificate** (brand Solvera, tanda tangan digital, watermark).

8) Otomasi (Cron, Server Action, Sequence)

Sequence:

- `seq_ojt_batch`: `OJTBATCH/%(year)s/%(seq)05d`
- `seq_ojt_certificate`: `OJTCERT/%(year)s/%(seq)06d`

Cron:

- `ojt_attendance_autoclose`: set `absent` jika lewat X menit tanpa check-in.
- `ojt_assignment_deadline_reminder`: kirim reminder H-1, H-0.
- `ojt_batch_state_switcher`: auto switch `recruit→ongoing→done` berdasarkan tanggal/event.

Server Actions:

- On `hr.applicant` Accepted → create `ojt.participant` + email portal invite.
- On `ojt.batch` action “Generate Certificates”.

9) API/Controller Publik

9.1 Verifikasi Sertifikat (Publik)

Route GET: `/ojt/cert/verify`
Params: `serial` **atau** `token`
Output (HTML/JSON):

- `valid: True/False`
- `participant_name`, `batch_name`, `issued_on`, `grade`, `final_score`, `attendance_rate`
- Tautan unduh PDF (jika public)

Contoh (controller ringkas, pseudo-Python):

```
class OJTCertificateController(http.Controller):
    @http.route(['/ojt/cert/verify'], type='http', auth='public',
methods=['GET'])
    def verify(self, serial=None, token=None, **kw):
        Cert = request.env['ojt.certificate'].sudo()
        domain = []
        if serial:
            domain = [('serial', '=', serial), ('state', '=',
'issued')]
        elif token:
            domain = [('qr_token', '=', token), ('state', '=',
'issued')]
        cert = Cert.search(domain, limit=1)
        if not cert:
            return
request.render('solvera_ojt_core.template_cert_invalid', {})
        vals = {
            'participant': cert.partner_id.name,
            'batch': cert.batch_id.name,
            'issued_on': cert.issued_on,
            'grade': cert.grade,
            'final_score': cert.final_score,
            'attendance_rate': cert.attendance_rate,
            'serial': cert.serial,
        }
        return request.render('solvera_ojt_core.template_cert_valid',
vals)
```

9.2 QR Check-in (Publik/Portal)

Route GET: `/ojt/attend/checkin/<string:token>`

- Token unik event+participant (atau event umum → form input email).
- Mencatat `check_in` & menandai `presence`.

---

10) Validasi & Constraint Teknis

- **Batch overlap**: Izinkan peserta ikut beberapa batch? (default: ya, tapi per bisnis bisa dibatasi)
- **Duplicate attendance**: SQL constraint `(participant_id, event_id)` unique.
- **Skor final**: Pastikan semua bobot normal (Σ weight boleh != 1, tapi formula menangani).
- **File size limit**: Batasi ukuran `ir.attachment` sesuai kebijakan.

---

11) Integrasi Modul Standar

- **Recruitment** → **OJT**: Smart button di `hr.applicant` → “Enroll to OJT” (pilih `ojt.batch`).
- **Event**: Gunakan `event.event.event_type_id` spesifik OJT bila ingin memfilter.
- **eLearning (website_slides)**:
    - Enrollment `slide.channel` saat peserta aktif.
    - (Opsional) Ambil completion rate course untuk kontribusi `score_final`.
- **Survey**: `survey.user_input.quizz_score` → agregasi ke `score_final`.

---

12) UX Detail Penting

- **Batch Kanban**: card menampilkan:
    - Progress rata-rata (%), jumlah peserta aktif, agenda terdekat, status.
- **Participant Form**: smart buttons → *Assignments, Attendance, Certificate, Courses*.
- **Assignment Quick Scoring**: list submissions dengan inline `score` + `feedback` (x2 klik).
- **Portal Peserta**: progress bar besar + CTA “Join Next Session” (ambil dari agenda terdekat).

---

13) Testing/QA (High-Level)

- **Unit Tests**:
    - Generate participant from applicant.
    - Attendance compute: present/late/absent.
    - Assignment scoring & late flag.
    - Certificate eligibility & sequence issuance.
    - Controller verify (valid/invalid).

- **Performance**:
  - 500 peserta/batch: load dashboard < 2s (dengan indexing & compute stored).
- **Security**:
  - Portal hanya data sendiri.
  - Mentor hanya batch yang ditugaskan.

---

14) Data Migration & Seed

- Seed `ojt.batch.type` (jika butuh klasifikasi), contoh batch “OJT BA Oct-Nov 2025”.
- Template QWeb Certificate ber-brand Solvera (logo, warna, watermark).
- Email templates:

  - Portal invite
  - Assignment reminder
  - Certificate issued

---

15) Rumus & Compute (Ringkas)

- `ojt.participant.attendance_rate`
  = `present_or_late_sessions / mandatory_sessions * 100`

- `ojt.participant.score_final` (contoh)
  = `Σ(assignment_score/max_score * 100 * assignment.weight)`
  - `mentor_score * w_mentor`
  - `quiz_score * w_quiz`
    (Bobot didefinisikan di batch atau assignment; normalisasi sesuai kebutuhan.)

- `ojt.certificate.grade` (contoh default)
  - `>= 85` → A
  - `>= 75` → B
  - else → C

---

16) Artefak Teknis (Struktur Modul)

```
solvera_ojt_core/
├── __manifest__.py
├── __init__.py
├── security/
│   ├── ir.model.access.csv
│   └── ojt_security.xml        # groups & rules
├── models/
│   ├── __init__.py
│   ├── ojt_batch.py
│   ├── ojt_event_link.py
│   ├── ojt_participant.py
│   ├── ojt_assignment.py
│   ├── ojt_assignment_submit.py
│   ├── ojt_attendance.py
│   └── ojt_certificate.py
├── controllers/
│   └── portal_public.py         # verify cert, QR check-in
├── data/
│   ├── ir_sequence.xml
│   ├── mail_templates.xml
│   ├── cron.xml
│   └── server_actions.xml
├── views/
│   ├── menu.xml
│   ├── ojt_batch_views.xml
│   ├── ojt_event_link_views.xml
│   ├── ojt_participant_views.xml
│   ├── ojt_assignment_views.xml
│   ├── ojt_assignment_submit_views.xml
│   ├── ojt_attendance_views.xml
│   ├── ojt_certificate_views.xml
│   └── portal_templates.xml     # website/portal
├── report/
│   ├── certificate_templates.xml
│   └── report_actions.xml
└── static/
    └── description/ (icon, screenshots)
```

17) Contoh Potongan Kode (Singkat)

17.1 Compute `attendance_rate` (Participant)

```
class OJTParticipant(models.Model):
    _name = 'ojt.participant'
    _inherit = ['mail.thread', 'mail.activity.mixin']
    # fields ...

    attendance_rate = fields.Float(compute='_compute_kpi', store=True)
    score_avg = fields.Float(compute='_compute_kpi', store=True)
    score_final = fields.Float(compute='_compute_kpi', store=True)

    def _compute_kpi(self):
        for rec in self:
            atts =
self.env['ojt.attendance'].search([('participant_id','=',rec.id)])
            mand_links = rec.batch_id.event_link_ids.filtered(lambda
l: l.is_mandatory)
            total_mand = len(mand_links)
            present_cnt = len([a for a in atts if
a.event_link_id.is_mandatory and a.presence in ('present','late')])
            rec.attendance_rate = (present_cnt/total_mand*100.0) if
total_mand else 0.0

            subs =
self.env['ojt.assignment.submit'].search([('participant_id','=',rec.id
),('state','in',('submitted','scored'))])
            scores = []
            for s in subs:
                if s.assignment_id.max_score:
                    scores.append((s.score /
s.assignment_id.max_score) * 100.0 * s.assignment_id.weight)
            rec.score_avg = sum(scores)/len(scores) if scores else 0.0
            # TODO: tambah komponen mentor_score / survey_score sesuai kebijakan
            rec.score_final = rec.score_avg
```

17.2 Issue Certificate (Wizard/Action)

```
eligible = participants.filtered(
    lambda p: p.attendance_rate >= batch.certificate_rule_attendance and
              p.score_final >= batch.certificate_rule_score and
              p.state == 'completed'
)
for p in eligible:
    self.env['ojt.certificate'].create({
        'name': f'Certificate {batch.name}',
        'batch_id': batch.id,
        'participant_id': p.id,
        'serial': self.env['ir.sequence'].next_by_code('ojt.certificate'),
        'qr_token': str(uuid.uuid4()),
        'attendance_rate': p.attendance_rate,
        'final_score': p.score_final,
        'state': 'issued',
    })
```

---

18) KPI & Monitoring

- **Batch KPI:** peserta aktif, avg attendance, avg final score, pass rate, sertifikat terbit.
- **Mentor KPI:** jumlah tugas dinilai, SLA penilaian (hari).
- **Peserta KPI (portal):** progress%, tugas selesai, kehadiran%, skor rata-rata.

---

19) Roadmap Pengembangan (Opsional)

1. **Integrasi Meeting Attendance** (Graph API/Zoom/Meet) → auto check-in/out.

2. **Gamification** (badges/points dari `website_slides`).
3. **Proctoring Ringan** untuk kuis akhir.
4. **API Public v1** (JSON) untuk verifikasi & daftar lulusan.