JejakAnak - Dummy Web Prototype Master Plan

Executive Summary

Sebelum memulai development aplikasi Flutter final, kita akan membangun **dummy web application** sebagai high-fidelity interactive prototype. Dummy web app ini akan meniru fitur dan user interface dari aplikasi real menggunakan front-end code dan realistic placeholder data. Tujuan utama adalah **memvalidasi requirements dan design** dengan team dan stakeholders secara early, memastikan semua orang memiliki pemahaman yang clear tentang functionality sebelum heavy development dimulai.

Manfaat Utama:

- ✅ **Validate Requirements**: Confirm semua fitur yang direncanakan benar-benar dibutuhkan
- ✅ **Test UX Flow**: Identifikasi friction points dalam user journey
- ✅ **Stakeholder Alignment**: Psikolog, educator, dan parent tester dapat memberikan feedback konkret
- ✅ **Reduce Development Risk**: Catch issues sebelum invest banyak resource di coding
- ✅ **Accelerate Development**: Blueprint yang jelas untuk developer (reference design)

Timeline: 2-3 weeks (Week 1-3 dari project roadmap)

Deliverables:

1. Clickable web prototype dengan semua key screens
2. Realistic dummy data (30-50 activities, 3-5 sample children with logs)
3. Validation report dari psikolog team, Uwa Farah team, dan parent testers
4. Refined requirements dan design ready untuk development

---

1. Objectives & Success Criteria

1.1 Primary Objectives

Requirement Validation:

- Confirm ALL functional requirements dari masterplan implemented correctly
- Identify missing features atau edge cases yang belum terpikirkan
- Validate 4E framework implementation (apakah UI-nya jelas untuk parents)
- Ensure prenatal to teen coverage makes sense dalam praktik

UI/UX Validation:

- Test navigation flow (apakah intuitive?)
- Validate information architecture (apakah mudah find things?)
- Test onboarding experience (apakah clear untuk first-time users?)
- Ensure activity logging process tidak tedious (friction-free)
- Validate dashboard insights (apakah helpful atau confusing?)

Stakeholder Buy-In:

- Psikolog team confirms educational value dan developmental appropriateness
- Uwa Farah team confirms content structure works for their workflow (admin panel)
- Parent testers confirm app solves real problems dan would use regularly
- Tech team confident tentang feasibility

Design Refinement:

- Identify UI improvements (layout, colors, typography, iconography)
- Discover missing screens atau modals
- Refine copy/messaging (apakah relatable untuk target audience?)
- Optimize for mobile mental model (even though prototype is web)

1.2 Success Criteria

Prototype dianggap sukses jika:

- ✅ All key user flows dapat dilalui dari start to finish
- ✅ Psikolog team: "Yes, ini approach-nya educationally sound"
- ✅ Uwa Farah team: "Yes, kami understand how to input content"
- ✅ Parent testers (5 people): Rata-rata rating 4/5 atau higher untuk usability
- ✅ Zero critical confusions (no one says "I don't understand what this app does")
- ✅ Tech team: "We're confident we can build this"

1.3 Out of Scope (What Prototype Won't Do)

- ❌ **No Real Backend**: Tidak ada database, API, atau server logic. Semua data static/mock
- ❌ **No Real Authentication**: Login form exists tapi tidak actual validation (any input diterima)
- ❌ **No Actual Photo Upload**: Photo upload UI exists tapi tidak store anywhere (simulate dengan placeholder)
- ❌ **No Push Notifications**: Cannot test notification delivery
- ❌ **No Real Gamification Calculations**: XP, badges shown tapi tidak calculated dynamically
- ❌ **No Performance Testing**: Prototype tidak need to be fast, just functional
- ❌ **No Security**: Prototype tidak have secure authentication or HTTPS (local/staging only)

---

2. Prototype Scope: Key Screens & Flows

2.1 Screen Inventory (Priority Order)

MUST HAVE (Core flows - 15 screens):

1. **Splash Screen**: Logo + tagline
2. **Welcome/Onboarding**: 3 slide carousel (intro, features, benefits)
3. **Sign Up/Login**: Single screen with tabs
4. **Profile Setup**: Parent name, photo upload (mock)
5. **Add Child - Step 1**: Name, status (expecting/born), date picker
6. **Add Child - Step 2**: Photo, gender, notes
7. **Home Dashboard**: Main screen after login
   - Child selector (if multiple)
   - Recommended activities section
   - Scheduled activities
   - Quick stats
   - Recent memories
8. **Activity Library**: Browse/search view
   - Filter bar (age, category, duration)
   - Activity cards (grid or list)
9. **Activity Detail**: Full activity information
   - Hero image
   - Description, materials, steps
   - Educational value
   - Schedule/Mark as Done buttons
10. **Schedule Activity Modal**: Date/time picker, reminder options
11. **Log Activity - Step 1**: Photo upload interface (mockup)
12. **Log Activity - Step 2**: Notes textarea
13. **Log Activity - Step 3**: 4E feedback toggles
14. **Child Timeline**: Chronological list of logged activities
15. **Child Dashboard (Insights)**: Stats, charts, talent indicators

SHOULD HAVE (Important but can be simplified - 8 screens):

16. **Settings**: Notification preferences, account settings
17. **Edit Profile**: Parent and child profile editing
18. **Activity Schedule List**: Calendar view of scheduled activities
19. **Recommendations Explained**: Modal showing why activity was recommended
20. **Badge Showcase**: Grid of earned and locked badges
21. **Challenge Page**: Active and past challenges
22. **Play Spot Detail**: Map view + spot info (simplified)
23. **Help/FAQ**: Static content page

NICE TO HAVE (If time permits - 5 screens):

24. **Admin Panel - Dashboard**: Stats for content team
25. **Admin Panel - Create Activity**: Template-based form
26. **Admin Panel - Review Queue**: Pending submissions
27. **Activity Variations Display**: Other parents' contributions
28. **Recipe Exchange Tab**: User-submitted recipes

TOTAL: 15-28 screens (depending on time)

2.2 User Flow Maps

Flow A: First-Time User Onboarding

```
Splash (2s auto-advance)
  ↓
Welcome Screen
  ├─ Slide 1: "Dokumentasikan perjalanan si kecil"
  ├─ Slide 2: "Aktivitas edukatif ter-kurasi"
  └─ Slide 3: "Temukan bakat anak Anda"
  ↓
  [Get Started] button
  ↓
Sign Up Screen
  ├─ Email + Password fields
  ├─ OR Social login buttons (Google, Apple - mockup)
  └─ [Create Account] button
  ↓
Profile Setup
  ├─ "What's your name?"
  ├─ Upload photo (placeholder image)
  └─ [Next]
  ↓
Add Child - Step 1
  ├─ "Tell us about your child"
  ├─ Name input
  ├─ Status radio: ○ Expecting (EDD picker) / ○ Born (birth date picker)
  └─ [Next]
  ↓
Add Child - Step 2
  ├─ Photo upload (mockup)
  ├─ Gender dropdown (optional)
  ├─ Notes textarea (optional)
  └─ [Complete]
  ↓
Personalization Modal
  ├─ "Help us personalize"
  ├─ Notification preferences checkboxes
  └─ [Finish]
  ↓
Home Dashboard (with tooltip tour overlay - skippable)
```

Flow B: Browse & Schedule Activity

```
Home Dashboard
  ↓
Scroll to "Recommended for [Child]" section
  ↓
Tap activity card
  ↓
Activity Detail Page
  ├─ Swipe through images
  ├─ Read description, materials, steps
  ├─ Expand "Parent Tips"
  ├─ See "What They'll Learn"
  └─ Action buttons at bottom
  ↓
Tap [Schedule] button
  ↓
Schedule Modal
  ├─ Date picker (calendar widget)
  ├─ Time picker (optional)
  ├─ Reminder dropdown (15min/1hr/1day before)
  ├─ Notes input (optional)
  └─ [Confirm]
  ↓
Toast: "Activity scheduled! ✅"
  ↓
Return to Dashboard
  └─ New card in "Scheduled Activities" section
```

Flow C: Log Activity (Core Value Loop)

```
Home Dashboard OR Activity Detail
  ↓
Tap [Mark as Done] button
  ↓
Log Activity Screen - Step 1: Photos
  ├─ [Take Photo] button (simulated with file picker)
  ├─ [Choose from Gallery] button (file picker)
  ├─ Show selected thumbnails (max 10)
  ├─ Tap thumbnail → add caption modal
  └─ [Next] or [Skip]
  ↓
Log Activity Screen - Step 2: Notes
  ├─ "How did it go?"
  ├─ Textarea (placeholder prompts)
  ├─ Character count (0/500)
  └─ [Next] or [Skip]
  ↓
Log Activity Screen - Step 3: 4E Feedback
  ├─ 😊 Enjoy: 🟣 Yes / 🟣 Neutral / 🟣 No
  ├─ ✨ Easy: 🟣 Too Easy / 🟣 Just Right / 🟣 Challenging
  ├─ 🌟 Excellent: 🟣 Yes / 🟣 Okay / 🟣 Struggled
  └─ 💎 Earn: 🟣 Yes / 🟣 No
  ↓
 [Save Activity] button
  ↓
Success Animation
  ├─ Confetti effect (CSS animation)
  ├─ "+10 XP" badge animation
  ├─ Check if badge unlocked → Show badge modal
  └─ [View Timeline] or [Back Home]
  ↓
Timeline Updated
  └─ New entry card appears at top
```

Flow D: View Insights

```
Home Dashboard
 ↓
Tap on child stats card or [View Insights] button
 ↓
Child Dashboard (Insights Page)
  ├─ Header: Child photo + name + age
  ├─ Activity Breakdown Chart (pie chart - static image for prototype)
  ├─ Favorite Activities section (cards with "❤️ Loved" count)
  ├─ Talent Indicators section
  │   └─ Insight cards:
  │      "🎨 Rina shows creative talent!"
  │      "Based on 8 activities where she excelled"
  │      [Tap to see details modal]
  ├─ Areas to Explore
  │   └─ "Haven't tried Science yet!"
  │      └─ Carousel of 3 Science activity cards
  └─ [View Full Timeline] button
 ↓
Timeline View
  ├─ Vertical scroll
  ├─ Month/year headers
  ├─ Activity log cards (date, photo, title, category)
  └─ Tap card → Log detail modal
```

Flow E: Admin - Create Activity (Simplified)

```
Admin Login Page (separate URL: /admin)
  ├─ Email + Password
  └─ [Login] (any input accepted in prototype)
  ↓
Admin Dashboard
  ├─ Quick stats cards
  ├─ Recent activity feed
  └─ [Create New Activity] CTA
  ↓
Template Selector Modal
  ├─ 🎨 Art & Craft
  ├─ 🔬 Science
  ├─ 🏃 Physical
  ├─ 🍽️ Cooking
  └─ ... (8 options)
  ↓
Create Activity Form (Wizard)
  ├─ Step 1: Basic Info (title, age, categories, difficulty, duration)
  ├─ Step 2: Materials (list builder)
  ├─ Step 3: Instructions (steps with image upload mockup)
  ├─ Step 4: Educational Value (checklists, parent tips textarea)
  └─ Step 5: Media & Preview
      ├─ Upload main image (file picker)
      ├─ [Preview Activity] button → Opens mobile view modal
      └─ [Save as Draft] / [Submit for Review] / [Publish]
  ↓
Success Message
  └─ "Activity created! 🎉"
  ↓
Return to Admin Dashboard (new activity in list)
```

---

3. Dummy Data Strategy

3.1 Realistic Data Principles

Why Realistic Data Matters:

- Testers can understand context (lorem ipsum is distracting)
- Reveals content strategy issues (e.g., text too long breaks layout)
- Allows meaningful feedback ("This activity sounds fun!" vs "Lorem ipsum dolor sit")
- Demonstrates app's value proposition convincingly

Data Realism Guidelines:

- ✅ Use actual Indonesian names, places, contexts
- ✅ Age-appropriate activity titles and descriptions
- ✅ Realistic photos (use free stock photos: Unsplash, Pexels)
- ✅ Plausible timestamps (recent dates, not 1970)
- ✅ Varied data (different categories, difficulties, durations)
- ❌ No real personal data (use obviously fake emails like test@jejakanak.app)
- ❌ No child faces in photos (privacy, use illustrations or partial shots)

3.2 Activity Library Data (50 Activities Minimum)

Category Distribution:

- Art & Creativity: 8 activities
- Science & Discovery: 6 activities
- Physical & Outdoor: 8 activities
- Literacy & Language: 6 activities
- Music & Rhythm: 4 activities
- Cooking & Nutrition: 6 activities
- Social & Emotional: 6 activities
- Cognitive & Logic: 6 activities

Age Distribution:

- Prenatal (0 months): 3 activities
- 0-12 months: 5 activities
- 1-3 years: 10 activities
- 3-5 years: 12 activities
- 6-8 years: 10 activities
- 9-11 years: 6 activities
- 12-15 years: 4 activities

Sample Activities (Examples for prototype):

json

```json
{
  "id": "act-001",
  "title": "Membuat Kolase dari Daun Kering",
  "slug": "kolase-daun-kering",
  "description": "Ajak si kecil mengumpulkan daun kering dan membuat karya seni kolase yang indah. Aktivitas ini melatih kr
  "ageMinMonths": 36,
  "ageMaxMonths": 84,
  "categories": ["Art & Creativity", "Outdoor"],
  "difficulty": "easy",
  "durationMinutes": 30,
  "locationType": "both",
  "materials": [
    {"name": "Daun kering", "quantity": "10-15 lembar", "isCommon": true},
    {"name": "Lem putih", "quantity": "1 botol", "isCommon": true},
    {"name": "Kertas karton", "quantity": "1 lembar", "isCommon": true},
    {"name": "Spidol warna", "quantity": "Set", "isCommon": true}
  ],
  "steps": [
    {
      "order": 1,
      "instruction": "Ajak anak berjalan-jalan di taman atau halaman untuk mengumpulkan daun kering. Biarkan mereka memil
      "imageUrl": "/images/activities/leaf-collage-step1.jpg"
    },
    {
      "order": 2,
      "instruction": "Di rumah, tunjukkan cara mengoleskan lem di balik daun dengan jari atau kuas kecil.",
      "imageUrl": "/images/activities/leaf-collage-step2.jpg"
    },
    {
      "order": 3,
      "instruction": "Bantu anak menempelkan daun di kertas karton untuk membentuk gambar (misal: pohon, bunga, kupu-kup
      "imageUrl": null
    },
    {
      "order": 4,
      "instruction": "Setelah selesai, tambahkan detail dengan spidol warna. Biarkan karya mengering lalu pajang di dinding!",
      "imageUrl": null
    }
  ],
  "educationalValues": ["Creativity", "Fine Motor Skills", "Nature Appreciation", "Art Expression"],
  "parentTips": "Biarkan anak bereksplorasi dengan bentuk dan warna tanpa terlalu banyak instruksi. Tidak perlu 'sempurna' -
  "conversationStarters": [
    "Daun mana yang paling kamu suka? Kenapa?",
    "Kalau daun ini bisa bicara, kira-kira dia cerita apa?",
    "Apa yang kamu rasakan saat pegang daun kering?"
  ],
  "safetyNotes": "Pastikan daun yang dikumpulkan bersih dan tidak beracun. Hindari daun dengan duri atau getah. Awasi peng
  "mainImageUrl": "/images/activities/leaf-collage-main.jpg",
  "additionalImages": [
    "/images/activities/leaf-collage-2.jpg",
    "/images/activities/leaf-collage-3.jpg"
  ],
  "videoUrl": null,
  "popularityScore": 85,
  "viewCount": 342,
  "completionCount": 127,
  "averageRating": 4.7,
  "publishedAt": "2025-09-15T08:00:00Z"
}
```

More Examples (Titles Only for Speed):

1. "Eksperimen Gunung Berapi Mini" (Science, 5-10 years)
2. "Yoga untuk Bumil: Trimester 2" (Prenatal)
3. "Sensory Play: Kotak Harta Karun" (0-2 years)
4. "Membuat Smoothie Buah Bersama" (Cooking, 3-7 years)
5. "Berburu Bentuk di Sekitar Rumah" (Outdoor, 2-5 years)
6. "Cerita Bergambar Buatan Sendiri" (Literacy, 6-10 years)
7. "Lagu & Gerakan: 'Kepala Pundak Lutut Kaki'" (Music, 1-4 years)
8. "Puzzle DIY dari Gambar Keluarga" (Cognitive, 4-8 years)
9. "Berkebun Mini: Menanam Kacang Hijau" (Science/Outdoor, 5-10 years)
10. "Napas dalam untuk Bumil" (Prenatal) ... (40 more activities covering all categories and ages)

3.3 User & Child Profile Data (5 Sample Parents)

Parent 1: Siti Rahmawati

- Email: siti.test@jejakanak.app
- Level: 3 (Active Parent)
- XP: 550
- Current Streak: 5 days
- Badges: Early Bird, Memory Keeper, Art Explorer
- Children:
  - **Rina** (4 years old, born 2020-03-15)
    - 18 activities logged
    - Favorite categories: Art (8), Music (4), Outdoor (3)
    - Insights: "Shows creative talent", "Loves hands-on activities"

Parent 2: Budi Santoso

- Email: budi.test@jejakanak.app
- Level: 4 (Super Parent)
- XP: 1,245
- Current Streak: 12 days
- Badges: Early Bird, Memory Keeper, Well-Rounded, Week Warrior, Super Parent
- Children:
  - **Arya** (7 years old, born 2018-01-22)
    - 35 activities logged
    - Favorite categories: Science (12), Outdoor (10), Cognitive (8)
    - Insights: "Curious about how things work", "Ready for advanced challenges"
  - **Sari** (4 years old, born 2021-06-10)
    - 15 activities logged
    - Favorite categories: Cooking (6), Art (5), Social (4)
    - Insights: "Enjoys social interaction", "Budding chef!"

Parent 3: Dewi Lestari (Bumil)

- Email: dewi.test@jejakanak.app
- Level: 1 (Newbie Parent)
- XP: 45
- Current Streak: 0 days
- Badges: Early Bird
- Children:
  - **Baby Lestari** (Expecting, EDD: 2026-02-14)
    - 3 prenatal activities logged
    - Notes: "Feeling prepared and excited!"

Parent 4: Rudi Hartono

- Email: rudi.test@jejakanak.app
- Level: 2 (Engaged Parent)
- XP: 280
- Current Streak: 3 days
- Badges: Early Bird, Memory Keeper
- Children:
  - **Kevin** (10 years old, born 2015-08-05)
    - 12 activities logged
    - Favorite categories: Physical (5), Science (4), Outdoor (3)
    - Insights: "Loves active play", "Try team sports?"

Parent 5: Ani Wijaya

- Email: ani.test@jejakanak.app
- Level: 3 (Active Parent)
- XP: 620
- Current Streak: 0 days (streak broken yesterday)
- Badges: Early Bird, Memory Keeper, Art Explorer, Storyteller (10 detailed notes)
- Children:
  - **Luna** (5 years old, born 2020-11-30)
    - 22 activities logged
    - Favorite categories: Literacy (9), Art (7), Music (4)
    - Insights: "Loves stories and books", "Strong language development"

3.4 Activity Logs (Sample Data for Timelines)

For Rina (Siti's daughter) - Recent 5 Logs:

json

```
[
  {
    "id": "log-001",
    "activityTitle": "Membuat Kolase dari Daun Kering",
    "completedAt": "2025-10-18T15:30:00Z",
    "photos": [
      "/images/logs/rina-collage1.jpg",
      "/images/logs/rina-collage2.jpg"
    ],
    "notes": "Rina sangat excited mengumpulkan daun! Dia bikin gambar kupu-kupu dari daun oak. Warnanya cantik banget - e
    "feedback4E": {
      "enjoy": true,
      "easy": true,
      "excellent": true,
      "earn": true
    },
    "isFavorite": true
  },
  {
    "id": "log-002",
    "activityTitle": "Lagu & Gerakan: 'Kepala Pundak Lutut Kaki'",
    "completedAt": "2025-10-16T10:00:00Z",
    "photos": ["/images/logs/rina-song.jpg"],
    "notes": "Rina udah hafal lagunya! Sekarang dia yang lead, Ibu yang ikut 😄 Gerakan kakinya udah koordinasi banget.",
    "feedback4E": {
      "enjoy": true,
      "easy": false,
      "excellent": true,
      "earn": false
    },
    "isFavorite": false
  },
  {
    "id": "log-003",
    "activityTitle": "Berburu Bentuk di Sekitar Rumah",
    "completedAt": "2025-10-14T16:45:00Z",
    "photos": [],
    "notes": "Rina nemuin banyak bentuk lingkaran (jam dinding, piring) dan persegi (jendela, bantal). Aktivitas ini simple tapi
    "feedback4E": {
      "enjoy": false,
      "easy": true,
      "excellent": false,
      "earn": false
    },
    "isFavorite": fal
    "isFavorite": false
  },
  {
    "id": "log-004",
    "activityTitle": "Sensory Play: Kotak Harta Karun",
    "completedAt": "2025-10-12T14:20:00Z",
    "photos": ["/images/logs/rina-sensory.jpg"],
    "notes": "Bikin sensory bin pakai beras dan mainan kecil. Rina main 45 menit nonstop! Focused banget.",
    "feedback4E": {
      "enjoy": true,
      "easy": true,
      "excellent": false,
      "earn": false
    },
    "isFavorite": false
  },
  {
    "id": "log-005",
    "activityTitle": "Membuat Smoothie Buah Bersama",
    "completedAt": "2025-10-10T09:00:00Z",
    "photos": ["/images/logs/rina-smoothie1.jpg", "/images/logs/rina-smoothie2.jpg"],
    "notes": "Rina suka banget masukin buah ke blender! Smoothie pisang-strawberry jadi favorit. Bonus: dia mau minum bua
    "feedback4E": {
      "enjoy": true,
      "easy": true,
      "excellent": false,
      "earn": true
    },
    "isFavorite": true
  }
]
```

(Similar log data for other children - total ~80 logs across 5 parents)

3.5 Gamification Data

Badges (Pre-defined in System):

javascript

```javascript
const BADGES = [
  {
    type: "early_bird",
    name: "Early Bird",
    description: "Logged your first 5 activities",
    icon: "🌱",
    unlockedBy: ["siti", "budi", "dewi", "rudi", "ani"] // all have this
  },
  {
    type: "memory_keeper",
    name: "Memory Keeper",
    description: "Uploaded 20 photos",
    icon: "📷",
    unlockedBy: ["siti", "budi", "ani"]
  },
  {
    type: "art_explorer",
    name: "Art Explorer",
    description: "Completed 10 Art activities",
    icon: "🎨",
    unlockedBy: ["siti", "ani"]
  },
  {
    type: "week_warrior",
    name: "Week Warrior",
    description: "7-day activity streak",
    icon: "🔥",
    unlockedBy: ["budi"]
  },
  {
    type: "super_parent",
    name: "Super Parent",
    description: "Logged 50 activities",
    icon: "🏆",
    unlockedBy: ["budi"]
  },
  {
    type: "well_rounded",
    name: "Well-Rounded",
    description: "Tried all 8 categories",
    icon: "🌍",
    unlockedBy: ["budi"]
  },
  {
    type: "storyteller",
    name: "Storyteller",
    description: "Wrote 10 detailed notes (50+ characters)",
    icon: "✍️",
    unlockedBy: ["ani"]
  },
  // ... (15 more badges, most locked for all users)
];
```

Current Challenge (Active):

json

```
{
  "id": "challenge-oct-2025",
  "title": "October Outdoor Adventure",
  "description": "Complete 5 outdoor activities this month and earn the Outdoor Explorer badge!",
  "startDate": "2025-10-01",
  "endDate": "2025-10-31",
  "goalCount": 5,
  "category": "Outdoor",
  "participants": [
    {"userId": "siti", "progress": 3, "completed": false},
    {"userId": "budi", "progress": 5, "completed": true}, // completed!
    {"userId": "rudi", "progress": 2, "completed": false}
  ]
}
```

3.6 Community Data (If Included)

Play Spots (3 examples):

json

```json
[
  {
    "id": "spot-001",
    "name": "Taman Spathodea, BSD",
    "type": "playground",
    "address": "Jl. Pahlawan Seribu, BSD City, Tangerang Selatan",
    "latitude": -6.3018,
    "longitude": 106.6519,
    "ageSuitabilityMin": 1,
    "ageSuitabilityMax": 12,
    "facilities": ["parking", "restroom", "food_stall", "shade"],
    "photos": ["/images/playspots/spathodea.jpg"],
    "notes": "Great for toddlers! Ada ayunan, perosotan, dan area pasir. Pagi hari best time (sepi & sejuk).",
    "upvotes": 15,
    "activityCount": 8
  },
  {
    "id": "spot-002",
    "name": "Kidzania Jakarta",
    "type": "indoor_play",
    "address": "Pacific Place Mall, Jakarta",
    "latitude": -6.2252,
    "longitude": 106.8093,
    "ageSuitabilityMin": 4,
    "ageSuitabilityMax": 14,
    "facilities": ["parking", "restroom", "food_court", "AC"],
    "notes": "Role-play activities untuk anak. Lumayan pricey tapi worth it. Book online untuk avoid queue.",
    "upvotes": 23,
    "activityCount": 12
  },
  {
    "id": "spot-003",
    "name": "Perpustakaan Cikini",
    "type": "library",
    "address": "Jl. Cikini Raya, Jakarta Pusat",
    "ageSuitabilityMin": 3,
    "ageSuitabilityMax": 15,
    "facilities": ["parking", "restroom", "wheelchair_accessible"],
    "notes": "Koleksi buku anak bagus. Ada story telling session setiap Sabtu pagi jam 10. Free!",
    "upvotes": 9,
    "activityCount": 4
  }
]
```

Activity Variations (2 examples):

```json
[
  {
    "parentActivityTitle": "Membuat Kolase dari Daun Kering",
    "submittedBy": "ani",
    "variationTitle": "Luna's Underwater Scene Collage",
    "variationNotes": "Kami pakai daun kering untuk bikin pemandangan bawah laut! Daun oak jadi ubur-ubur, daun maple jac
    "photoUrl": "/images/variations/luna-underwater.jpg",
    "helpfulCount": 5
  },
  {
    "parentActivityTitle": "Membuat Smoothie Buah Bersama",
    "submittedBy": "siti",
    "variationTitle": "Green Monster Smoothie (Rina's Fave)",
    "variationNotes": "Tambah bayam + yogurt + madu. Rina surprisingly suka! Dia sebut 'jus monster hijau' 🤣 Healthy & yu
    "photoUrl": "/images/variations/green-smoothie.jpg",
    "helpfulCount": 12
  }
]
```

---

4. Technology Stack untuk Prototype

4.1 Evaluation Criteria

Prototype tech stack harus:

- ✅ **Fast to develop** (2-3 weeks deadline)
- ✅ **Easy to iterate** (quick changes based on feedback)
- ✅ **Realistic feel** (interactive, not just static mockup)
- ✅ **Accessible** (anyone can open and test, no complex setup)
- ✅ **Disposable** (tidak perlu production-quality code, akan di-rewrite di Flutter)

4.2 Recommended Stack

Option A: React + TailwindCSS (Recommended)

Pros:

- Component-based (easy to reuse elements)
- TailwindCSS untuk rapid styling (utility classes)
- React hooks untuk simple state management
- Banyak UI libraries gratis (Headless UI, Radix UI)
- Hot reload (instant preview)
- Team familiar (you know Vue, React similar)

Cons:

- Slight learning curve jika belum pernah React (but similar to Vue)
- Need Node.js setup

Tech Details:

- **Framework**: React 18 (via Vite for fast dev server)
- **Styling**: TailwindCSS + Headless UI (for modals, dropdowns)
- **Routing**: React Router (for multi-page navigation)
- **State**: React Context atau useState (simple, no Redux needed)
- **Icons**: Lucide React atau Heroicons
- **Charts**: Recharts (for dashboard pie chart)
- **Animations**: Framer Motion (for page transitions, confetti)
- **Date Picker**: React DatePicker

Setup Time: ~1 hour (create-react-app atau Vite)

Option B: Vue 3 + TailwindCSS (Alternative if Prefer Vue)

Pros:

- You're already familiar with Vue!
- Component-based
- TailwindCSS sama aja
- Composition API powerful

Cons:

- Slightly less library options vs React

Tech Details: Similar to React but with Vue ecosystem

Option C: Plain HTML + Bootstrap + Vanilla JS (Fastest but Less Elegant)

Pros:

- Zero setup (just open index.html in browser)
- No build process
- Anyone can edit (even non-developers)

Cons:

- Code becomes messy quickly (hard to maintain)
- Less interactive feel (more work for dynamic behavior)
- No component reusability

When to Use: Only if extreme time constraint (1 week instead of 2-3)

RECOMMENDATION: **React + TailwindCSS** (Option A)

Why: Sweet spot antara speed dan quality. Component reusability saves time. Tailwind makes styling fast. React's ecosystem rich (easy find copy-paste components).

4.3 Project Structure (React)

```
jejakanak-prototype/
├── public/
│   ├── index.html
│   ├── images/
│   │   ├── activities/      # Activity photos
│   │   ├── logs/            # User log photos
│   │   ├── playspots/       # Play spot photos
│   │   ├── avatars/         # Child avatars
│   │   └── badges/          # Badge icons
│   └── data/
│      ├── activities.json  # All 50 activities
│      ├── users.json       # 5 sample parents
│      ├── children.json    # Child profiles
│      ├── logs.json        # Activity logs
│      ├── badges.json      # Badge definitions
│      ├── challenges.json  # Active challenges
│      └── playspots.json   # Play spot data
├── src/
│   ├── components/
│   │   ├── ActivityCard.jsx
│   │   ├── ActivityDetail.jsx
│   │   ├── ChildCard.jsx
│   │   ├── LogCard.jsx
│   │   ├── BadgeIcon.jsx
│   │   ├── Modal.jsx
│   │   ├── Navbar.jsx
│   │   ├── BottomNav.jsx
│   │   └── ... (20-30 components)
│   ├── pages/
│   │   ├── Splash.jsx
│   │   ├── Welcome.jsx
│   │   ├── SignUp.jsx
│   │   ├── OnboardingAddChild.jsx
│   │   ├── Home.jsx
│   │   ├── ActivityLibrary.jsx
│   │   ├── ActivityDetailPage.jsx
│   │   ├── LogActivity.jsx
│   │   ├── ChildTimeline.jsx
│   │   ├── ChildDashboard.jsx
│   │   ├── Settings.jsx
│   │   └── ... (15-20 pages)
│   ├── context/
│   │   └── AppContext.jsx  # Global state (current user, child)
│   ├── hooks/
│   │   └── useLocalStorage.jsx  # Persist logged in user
│   ├── utils/
│   │   ├── calculateAge.js
│   │   ├── filterActivities.js
│   │   └── formatDate.js
│   ├── App.jsx          # Root component with Router
│   ├── index.css        # Tailwind imports
│   └── main.jsx         # Entry point
├── package.json
├── tailwind.config.js
├── vite.config.js
└── README.md
```

4.4 Data Management Strategy

Static JSON Files: All dummy data stored in `/public/data/` as JSON

Why JSON Files:

- No database needed (zero setup)
- Easy to edit (just open file, change data)
- Fast to load (fetch() from public folder)
- Version controllable (git tracks changes)

Data Flow:

```javascript
// In component
useEffect(() => {
  fetch('/data/activities.json')
    .then(res => res.json())
    .then(data => setActivities(data));
}, []);
```

State Management:

- **Global State** (Context API): Current logged-in user, selected child
- **Local State** (useState): Page-specific data (form inputs, filters)
- **Persistence** (localStorage): Remember logged-in user across sessions (simulate auth)

Example Context:

```javascript
const AppContext = createContext();

export function AppProvider({ children }) {
  const [currentUser, setCurrentUser] = useState(() => {
    // Load from localStorage if exists
    const saved = localStorage.getItem('currentUser');
    return saved ? JSON.parse(saved) : null;
  });

  const [selectedChild, setSelectedChild] = useState(null);

  const login = (email) => {
    // Simulate login - find user in data
    fetch('/data/users.json')
      .then(res => res.json())
      .then(users => {
        const user = users.find(u => u.email === email);
        setCurrentUser(user);
        localStorage.setItem('currentUser', JSON.stringify(user));
      });
  };

  return (
    <AppContext.Provider value={{
      currentUser,
      selectedChild,
      setSelectedChild,
      login
    }}>
      {children}
    </AppContext.Provider>
  );
}
```

4.5 Simulating Interactivity

Login/Signup:

- Any email input → Accepts
- Password field → Ignored (no validation)
- On submit → Find user in users.json by email (or create mock user)
- Store in context + localStorage

Adding Child:

- Form inputs → Store in local state
- On submit → Add to `currentUser.children` array in memory (not persisted)
- Navigate to home with new child

Logging Activity:

- Photo upload → File picker, store File objects in state, display thumbnails
- On save → Create log object, push to `selectedChild.logs` array in memory
- Show success animation, update timeline

Scheduling:

- Date picker → Store selected date in state
- On confirm → Add to `scheduledActivities` array in memory
- Display in "Upcoming" section

Gamification:

- XP/Badges → Pre-calculated in dummy data
- On log activity → Show "+10 XP" animation (not actually calculated)
- If new badge unlocked (pre-defined) → Show badge modal

Recommendations:

- Algorithm → Simplified (show predefined list for each child)
- "Recommended" activities → Pre-selected in data based on child's past logs

Charts:

- Dashboard pie chart → Static image or use Recharts with dummy percentages
- Data doesn't update dynamically (just for show)

Filters/Search:

- Activity library filters → Client-side filtering of activities.json
- Real-time search → Filter array by title match

Admin Panel:

- Create activity form → All fields functional
- On submit → Show success message, mock add to library (not persisted)
- Template selector → Pre-fill form with template data

---

5. Development Process (Week-by-Week)

Week 1: Foundation & Core Flows

Day 1-2: Setup & Data Preparation

□ Initialize React + Vite project
□ Install dependencies (Tailwind, React Router, etc.)
□ Configure Tailwind (colors, fonts matching brand)
□ Create project structure (folders, base components)
□ Prepare all dummy data JSON files (use GPT to generate if needed)
□ Gather images:

- Download 50+ free stock photos (Unsplash: kids, activities, parenting)
- Create placeholder avatars (Boring Avatars generator)
- Badge icons (emoji or simple SVGs)

Day 3-4: User Authentication & Onboarding

□ Build Splash screen (auto-advance after 2s)
□ Build Welcome screen (3-slide carousel)
□ Build SignUp/Login page
□ Implement Context for global state
□ Build Add Child flow (Step 1 & 2)
□ Build Personalization modal
□ Test full onboarding flow (Splash → Welcome → Sign Up → Add Child → Home)

Day 5-7: Home Dashboard & Activity Library

☐ Build Home Dashboard:

- Child selector (if multiple)
- Recommended activities section (cards)
- Scheduled activities section
- Quick stats cards
- Recent memories section

☐ Build Activity Library page:

- Grid of activity cards
- Filter bar (category, age, duration dropdowns)
- Search input (real-time filter)

☐ Build ActivityCard component (reusable)

☐ Build Activity Detail page:

- Image carousel
- Full content display (materials, steps, tips)
- Schedule button (opens modal)
- Mark as Done button

☐ Build Schedule modal (date/time pickers, reminder dropdown)

☐ Test flow: Home → Browse → Activity Detail → Schedule → Back to Home

---

Week 2: Core Value Loop & Insights

Day 8-10: Activity Logging

☐ Build Log Activity page (multi-step wizard):

- Step 1: Photo upload UI (file picker, thumbnails)
- Step 2: Notes textarea
- Step 3: 4E feedback (visual toggles)
- Navigation: Back/Next buttons, progress indicator

☐ Build success animation (confetti + XP badge)

☐ Build badge unlock modal (if applicable)

☐ Implement adding log to child's timeline (in-memory)

☐ Test flow: Activity Detail → Log → Success → Timeline updated

Day 11-12: Timeline & Insights Dashboard

□ Build Child Timeline page:

- Vertical scroll
- Month/year headers
- Log cards (date, photo, title, excerpt)
- Tap card → Expand detail modal

□ Build Child Dashboard (Insights) page:

- Header with child info
- Activity breakdown chart (pie chart - Recharts or static image)
- Favorite activities carousel
- Talent indicators section (insight cards)
- Areas to explore section (recommended categories)

□ Implement basic "insights" logic (count categories, detect favorites)

□ Test flow: Home → Child Dashboard → View insights → Explore recommended activities

Day 13-14: Polish Core Features

□ Build Settings page (notification preferences, account info)

□ Build Edit Profile pages (parent, child)

□ Implement bottom navigation (Home, Library, Add/Log, Timeline, Profile)

□ Add page transitions (smooth animations)

□ Add loading states (skeletons, spinners)

□ Add empty states ("No activities logged yet" with CTA)

□ Bug fixes and responsive tweaks (test on mobile size)

Week 3: Secondary Features & Admin Panel

Day 15-16: Gamification Elements

□ Build Badges page (grid of earned + locked badges)

□ Build Challenge page (active challenge with progress bar)

□ Add XP/level display to Home (user avatar with level badge)

□ Add streak indicator to dashboard ("🔥 5-day streak!")

□ Test badge animations and challenge progress

Day 17-18: Community Features (If Time Permits)

□ Build Play Spots page:
- Map view (embedded Google Maps iframe or static map image with pins)
- List view (spot cards with distance, upvotes)

□ Build Play Spot Detail modal

□ Build Activity Variations section (on activity detail page)
- Carousel of user variations
- "Helpful" button

□ Test community flows

Day 19-20: Admin Panel

□ Build Admin Login page (separate URL: `/admin`)

□ Build Admin Dashboard:
- Stats cards
- Recent activity feed
- Create Activity CTA

□ Build Create Activity page:
- Template selector modal
- Multi-step form wizard (5 steps)
- Preview modal (mobile view simulation)

□ Build Activity Library management table (list with edit/delete)

□ Test admin flows: Login → Dashboard → Create Activity → Preview → Save

Day 21: Final Polish & Testing

□ Cross-browser testing (Chrome, Safari, Firefox)

□ Mobile responsive testing (iPhone, Android sizes)

□ Fix any layout bugs

□ Add tooltips/help text where needed

□ Proofread all copy (check for typos)

□ Create a "demo account" with rich data for testers

□ Deploy prototype to Vercel/Netlify (free hosting)
- Get shareable URL: https://jejakanak-prototype.vercel.app

---

6. User Testing Protocol

6.1 Participant Recruitment

Target Participants:

Group A: Psikolog/Educator Team (2-3 people)

- Uwa Farah (lead validator)
- 1-2 child development experts
- Focus: Educational value, developmental appropriateness, safety

Group B: Content Team (2-3 people)

- Uwa Farah's content creators
- Focus: Admin panel usability, content input workflow

Group C: Parent Testers (5-7 people)

- Mix of demographics:
  - 2 first-time parents (young children 0-3)
  - 2 experienced parents (children 4-8)
  - 1 parent with multiple kids
  - 1 expectant mother (pregnant)
  - 1 parent with older child (9-15)
- Mix of tech-savviness (not all early adopters)
- All from target market (urban Indonesia)

Recruitment:

- Internal network (friends, colleagues)
- Parenting Facebook groups (post invitation)
- Incentive: Early access to app + "Founding Member" badge when launches

6.2 Testing Sessions

Format: Individual 1-hour sessions (remote via Zoom or in-person)

Session Structure:

Part 1: Introduction (5 min)

- Explain purpose: "Testing prototype app, not testing you"
- Emphasize: "Think aloud - say what you're thinking"
- Clarify: "This is dummy, some things won't work perfectly"
- Get consent (record session for note-taking)

Part 2: Guided Tasks (40 min)

Facilitate (but don't help), observe, take notes:

Task 1: Onboarding (10 min)

- "Imagine you just downloaded JejakAnak. Explore and sign up."
- Observe:
  - Do they understand value proposition from welcome screen?
  - Any confusion in sign up flow?
  - Add child process smooth?
  - Do they understand prenatal option?

Task 2: Browse Activities (10 min)

- "You want to find a fun art activity for your 4-year-old. Show me how you'd do that."
- Observe:
  - Can they find library easily?
  - Do they use filters or scroll?
  - Is activity detail page clear?
  - Do they understand materials/steps?

Task 3: Schedule Activity (5 min)

- "You found an activity you like. Schedule it for this weekend."
- Observe:
  - Can they find schedule button?
  - Is scheduling process intuitive?
  - Do they set reminder?

Task 4: Log Activity (10 min)

- "Pretend you just did this activity with your child. Log it."
- Observe:
  - Photo upload: confusing?
  - Notes: do they write something meaningful?
  - 4E feedback: do they understand each dimension?
  - Any friction?

Task 5: Explore Insights (5 min)

- "Check your child's progress and insights."
- Observe:
  - Can they find dashboard?
  - Do they understand insights?
  - Is chart readable?
  - Do recommendations make sense?

Part 3: Open Exploration (10 min)

- "Explore the app freely. Click around."
- See what they naturally gravitate to
- Note: What do they click? What do they ignore?

Part 4: Feedback Discussion (5 min)

- Structured questions:
  - "What did you like most?"
  - "What was confusing or frustrating?"
  - "Would you use this app? Why or why not?"
  - "What's missing?"
  - "On a scale 1-5, how likely to recommend?"
- Open-ended: "Any other thoughts?"

6.3 Psikolog Team Validation (Separate Session)

Format: 2-hour workshop

Focus Areas:

1. Educational Alignment (45 min)

- Review sample activities (10-15 activities)
- Questions:
  - Are age ranges appropriate?
  - Is educational value clearly articulated?
  - Any safety concerns?
  - Are conversation starters effective?
  - Anything developmentally inappropriate?

2. 4E Framework Validation (30 min)

- Show 4E feedback UI
- Questions:
  - Is this framework sound for talent discovery?
  - Are the 4 dimensions clear to parents?
  - Any concerns about misinterpretation?
  - Suggestions for improvement?

3. Insights Review (30 min)

- Show sample insights ("Rina shows creative talent")
- Questions:
  - Are these insights helpful or misleading?
  - Risk of labeling children too early?
  - How to phrase insights sensitively?
  - What insights are most valuable for parents?

4. Overall App Assessment (15 min)

- Would you recommend this to parents?
- Any red flags from child development perspective?
- Suggestions for improvement?

6.4 Content Team Validation (Admin Panel Focus)

Format: 1.5-hour hands-on session

Tasks:

Task 1: Create Activity (45 min)

- "Create a new art activity for 3-5 year olds using the admin panel."
- Observe:
  - Template helpful or restrictive?
  - Any confusing fields?
  - Validation errors helpful?
  - Preview function useful?

Task 2: Bulk Import (15 min)

- "You have 10 activities in a spreadsheet. How would you add them?"
- Show bulk import feature
- Get feedback on format and process

Task 3: Review Content (15 min)

- Show review queue with pending recipes/variations
- "Approve or reject these 3 submissions."
- Observe decision-making process

Discussion (15 min):

- Is this workflow efficient for your needs?
- What would make content creation easier?
- Any features missing?

6.5 Data Collection & Analysis

During Sessions:

- Screen recording (with permission)
- Observer notes (what worked, what didn't)
- Quotes (especially pain points or delights)
- Task completion: Yes/No + Time taken

After Sessions:

- Compile notes in shared doc (Notion or Google Doc)
- Categorize feedback:
    - **Critical**: Must fix before development (e.g., "Totally confused by 4E")
    - **High**: Should fix (e.g., "Schedule button hard to find")
    - **Medium**: Nice to have (e.g., "Would love dark mode")
    - **Low**: Future consideration (e.g., "Calendar view would be cool")

Metrics to Track:

- Task success rate (% who completed without help)
- Average time per task
- Number of errors/wrong clicks
- SUS Score (System Usability Scale - optional survey)
- NPS (Net Promoter Score): "Likelihood to recommend (0-10)"

Synthesis:

- Identify patterns (if 3+ people have same issue → critical)
- Prioritize fixes for iteration
- Create action items with owners

---

7. Iteration & Refinement

7.1 Feedback Synthesis Meeting (Day 22)

Attendees: Full team (product lead, tech lead, designer, Uwa Farah)

Agenda:

1. **Share key findings** (30 min):
   - Facilitator presents summary of user testing
   - Highlight critical issues, repeated feedback, positive reactions
   - Show video clips of key moments (confusion, delight)
2. **Prioritization exercise** (30 min):
   - Group feedback into Must-Fix / Should-Fix / Nice-to-Have
   - Estimate effort for each fix (1hr / half-day / 1 day)
   - Decide: What changes before development? What defers to later?
3. **Assign action items** (15 min):
   - Who fixes what?
   - Deadline: 2-3 days for changes
4. **Re-validation plan** (15 min):
   - Do we need another round of testing?
   - Or just stakeholder sign-off after fixes?

7.2 Common Issues & Solutions (Hypothetical)

Issue 1: "4E Feedback is confusing"

- **Finding**: 3/5 parents unsure what "Earn" means
- **Solution**: Change wording to "Create" or add tooltip explanation
- **Effort**: 1 hour (update copy + add tooltip)

Issue 2: "Too many steps to log activity"

- **Finding**: Parents feel 3-step wizard is long
- **Solution**: Make photo upload optional, allow skip all steps except save
- **Effort**: 2 hours (adjust validation)

Issue 3: "Can't find scheduled activities"

- **Finding**: Users look for calendar icon but there's only a list
- **Solution**: Add calendar icon to navbar, more prominent "Upcoming" section
- **Effort**: 3 hours (new icon, reorder dashboard)

Issue 4: "Admin form too long"

- **Finding**: Uwa Farah team overwhelmed by 5-step wizard
- **Solution**: Collapse optional sections, make wizard progress clearer
- **Effort**: 4 hours (UI refactor)

Issue 5: "Love the gamification!"

- **Finding**: All parents excited about badges and streaks
- **Solution**: No change needed, validate this works! Maybe add more badges.
- **Effort**: 0 hours (keep as is)

7.3 Iteration Sprint (Day 23-25)

Day 23-24: Implement high-priority fixes

- Focus on critical usability issues
- Update copy where confusing
- Adjust layouts for clarity

Day 25: Final polish

- One more round of internal testing
- Fix any new bugs introduced
- Update documentation (if any screens changed significantly)

7.4 Final Approval (Day 26)

Approval Meeting (1 hour):

- Present updated prototype to stakeholders
- Walkthrough changes made based on feedback
- Get sign-off from:
  - Uwa Farah (content perspective)
  - Psikolog team (educational validation)
  - Product owner (requirements met)
  - Tech lead (feasible to implement)

Deliverables:

- ✅ Approved prototype (locked version)
- ✅ Validation report (summary of findings + changes made)
- ✅ Design spec exported from prototype (screenshots + annotations for developers)
- ✅ Dummy data files (ready to use as seed data in real app)

---

8. Handoff to Development

8.1 Documentation Package

For Developers:

1. **Prototype Access**:
   - Live URL: https://jejakanak-prototype.vercel.app
   - Source code: GitHub repo (if open to team)
   - Login credentials: demo@jejakanak.app / password123
2. **Screen Specifications**:
   - Figma file (if exists) with final designs
   - OR: Exported screenshots from prototype with annotations
   - Spacing, colors, fonts documented (design system)
3. **User Flows**:
   - Flow diagrams (from Section 2.2 of this doc)
   - Critical paths highlighted
4. **Data Models**:
   - JSON schema for each entity (activity, user, child, log)
   - Relationships documented
   - Sample data files (activities.json, users.json, etc.)
5. **Functional Requirements**:
   - Feature list with detailed descriptions
   - Edge cases identified during testing
   - Known limitations of prototype (to be built properly in app)
6. **Testing Insights**:
   - Validation report summary
   - What worked well (keep as is)
   - What was confusing (pay extra attention)

8.2 Development Kickoff Meeting

Before Sprint 1 (Week 4 Day 1):

Agenda (1.5 hours):

1. **Prototype Walkthrough** (30 min):
   - Product lead demos prototype live
   - Highlights key features and flows
   - Explains why certain design decisions were made
2. **Technical Architecture Overview** (30 min):
   - Tech lead presents high-level architecture (Flutter + NestJS + Postgres)
   - Discusses how prototype maps to real implementation
   - Identifies technical challenges
3. **Sprint 1 Planning** (30 min):
   - Review Sprint 1 backlog (from masterplan)
   - Assign tasks to developers (or AI coding sessions)
   - Set Sprint 1 goal: "Working auth + child profile + basic activity browse"

Outputs:

- ✅ Development team understands requirements
- ✅ Sprint 1 tasks estimated and assigned
- ✅ Any blocking questions answered

8.3 Prototype as Living Reference

Throughout Development:

- Prototype stays deployed (URL bookmarked by all)
- Developers refer to prototype when unclear about UX
- QA testers use prototype as reference ("Does real app match prototype?")
- Product owner uses prototype to answer stakeholder questions

Sunset Plan:

- Prototype remains live until V1.0 launches
- After launch, archive prototype (keep GitHub repo for reference)

---

9. Success Metrics for Prototype Phase

9.1 Process Metrics

- ✅ **Timeline Met**: Prototype completed in 2-3 weeks
- ✅ **Budget**: No cost (free tools + free hosting)
- ✅ **Coverage**: All 15 must-have screens built
- ✅ **Testing**: 5+ parent testers + psikolog team + content team validated

9.2 Quality Metrics

- ✅ **Usability**: Average task success rate > 80%
- ✅ **Satisfaction**: Average user rating > 4/5
- ✅ **NPS**: Average Net Promoter Score > 40 (good)
- ✅ **Clarity**: Zero critical confusions in testing
- ✅ **Approval**: All stakeholders sign off

9.3 Outcome Metrics

- ✅ **Requirements Refined**: 10+ adjustments made based on feedback
- ✅ **Risk Reduced**: Major UX issues caught early (not during development)
- ✅ **Alignment**: Team confident in what to build
- ✅ **Accelerated Development**: Developers have clear blueprint

---

10. Risks & Mitigation

10.1 Prototype-Specific Risks

Risk: Prototype takes longer than 3 weeks

- **Mitigation**: Timebox strictly, cut nice-to-have screens if needed
- **Contingency**: Launch with 12 screens instead of 28, focus on core flows

Risk: User testing reveals fundamental flaws (concept doesn't resonate)

- **Mitigation**: Test early with 1-2 parents before building all screens
- **Contingency**: Pivot concept, rebuild prototype with new approach (adds 1-2 weeks)

Risk: Psikolog team rejects educational approach

- **Mitigation**: Involve them early in ideation, not just validation
- **Contingency**: Revise 4E framework or activity structure, iterate prototype

Risk: Technical implementation is way harder than prototype suggests

- **Mitigation**: Tech lead reviews prototype feasibility during build
- **Contingency**: Simplify certain features (e.g., less dynamic recommendations)

Risk: Stakeholder disagreement on changes after testing

- **Mitigation**: Clear decision-making framework (data-driven, product owner final say)
- **Contingency**: Escalate to senior leadership if deadlock

10.2 Resource Risks

Risk: Designer unavailable (delays visual design)

- **Mitigation**: Use pre-made Tailwind UI components, focus on functionality over aesthetics
- **Contingency**: Developer does basic design, iterate visuals later

Risk: Not enough realistic photos for dummy data

- **Mitigation**: Use Unsplash API (free high-quality photos)
- **Contingency**: Use illustrations or simple colored placeholders

---

11. Post-Prototype Action Items

11.1 Immediate Next Steps (After Approval)

Week 4 (Development Start):

1. **Archive Prototype** (1 hour):
   - Tag version in GitHub: `v1.0-validated`
   - Export all screens as PDF (documentation)
   - Update README with findings summary
2. **Extract Reusable Assets** (2 hours):
   - Export color palette (Tailwind config → Flutter theme)
   - Document typography scale (font sizes, weights)
   - Save icon set (export SVGs)
   - Collect dummy data files (use as seed data)
3. **Create Developer Handoff Doc** (3 hours):
   - Screen-by-screen specifications
   - Interaction notes ("On tap X, show modal Y")
   - Copy all final text/copy (for consistency)
   - List of edge cases discovered
4. **Setup Development Environment** (Per Masterplan):
   - Initialize Flutter project
   - Initialize NestJS backend
   - Setup database schema
   - Configure CI/CD pipeline

11.2 Ongoing Prototype Maintenance

During Development (Week 4-16):

- Keep prototype live for reference
- If major design changes during dev, optionally update prototype (low priority)
- Use prototype for demos to investors/partners if needed

After V1.0 Launch:

- Archive prototype (no longer maintained)
- Keep GitHub repo as historical reference
- Document lessons learned in team wiki

---

12. Budget & Resources

12.1 Cost Breakdown

Development Costs:

- Developer time: 2-3 weeks × 1 developer = **~120-180 hours**
  - If outsourced: Rp 150,000/hr × 150hr = Rp 22,500,000 (~$1,500 USD)
  - If internal: Opportunity cost only
- Designer time (if separate): ~20 hours for mockups
  - If outsourced: Rp 200,000/hr × 20hr = Rp 4,000,000 (~$270 USD)

Infrastructure Costs:

- Hosting (Vercel/Netlify): **Free** (for prototype scale)
- Domain (optional): Rp 150,000/year (~$10 USD)
- Stock photos: **Free** (Unsplash, Pexels)
- Design tools: **Free** (Figma free tier)

Testing Costs:

- User incentives: Rp 100,000/tester × 10 testers = Rp 1,000,000 (~$65 USD)
  - Or: Early access badge (no cash cost)

Total Estimated Cost: **Rp 0 - 27,500,000** (~$0-1,800 USD)

- Minimum (DIY, no outsourcing): Rp 0
- Maximum (outsource dev + design): Rp 27,500,000

RECOMMENDATION: Keep costs minimal (DIY), invest budget in actual app development instead

12.2 Team Allocation

Required Roles:

1. Developer/Builder (1 person, full-time 2-3 weeks):

- Skills: React/Vue, basic design sense
- Responsibilities: Build all screens, implement interactions, deploy
- Ideal candidate: You (familiar with Vue, can learn React quickly)

2. Content Preparer (1 person, part-time ~20 hours):

- Skills: Writing, child development knowledge
- Responsibilities: Create 50 activity descriptions, write copy
- Ideal candidate: Uwa Farah or team member

3. Design Consultant (1 person, part-time ~10 hours):

- Skills: UI/UX design
- Responsibilities: Create color scheme, suggest layouts, review screens
- Ideal candidate: Freelance designer or use AI tools (v0.dev, Galileo AI)
- **Alternative**: Skip this role, use Tailwind UI templates

4. Validator/Tester (1 person, part-time ~10 hours):

- Skills: QA mindset, attention to detail
- Responsibilities: Test all flows, document bugs, coordinate user testing
- Ideal candidate: Product owner or tech lead

Optional:

- **Copywriter**: For polished marketing copy (welcome screen, etc.)
- **Photographer**: For custom photos (or use stock)

Minimum Team: Just 1-2 people can build entire prototype (developer + content preparer)

---

13. Alternative: Low-Code Prototype Options

13.1 If Time is Extremely Constrained (1 Week Only)

Option: Figma Interactive Prototype

Pros:

- No coding (pure design tool)
- Very fast (1 week achievable)
- Professional-looking
- Easy to iterate (just edit design)

Cons:

- Less realistic (no real data handling)
- Limited interactivity (only link hotspots)
- Harder to simulate complex flows (multi-step forms)

When to Use: If team has no developers available and need validation ASAP

How:

1. Design all screens in Figma (use UI kits for speed)
2. Add interactive links (tap button → go to next screen)
3. Create multiple "paths" for different scenarios
4. Present mode = prototype
5. Share link with testers

Effort: ~40 hours (1 week)

Option: No-Code Builders (Bubble, Webflow, Softr)

Pros:

- Visual builder (drag-and-drop)
- Can handle data (better than Figma)
- Faster than code (for simple apps)

Cons:

- Learning curve (need to learn the tool)
- Limitations (complex interactions harder)
- Not free (most require subscription)
- Vendor lock-in (can't export code easily)

When to Use: If team wants semi-functional prototype with basic data handling but no coding

Recommendation: **Not recommended** for JejakAnak prototype

- Why: Mobile app UX hard to replicate in no-code builders
- Learning new tool takes time (defeats speed purpose)
- Code-based prototype more flexible for iterations

13.2 Hybrid Approach (Recommended for Time Savings)

Strategy: Design in Figma, Build Core Flows in Code

Week 1: Design all screens in Figma (fast, visual)

- Day 1-3: Design 20 screens
- Day 4-5: Interactive prototype in Figma (clickable)
- Day 5: Present to team for early feedback

Week 2: Build critical flows in code (realistic)

- Only build: Onboarding, Browse, Log Activity (3 key flows)
- Skip: Settings, Admin, secondary features
- Use Figma as reference for styling

Week 3: User testing + iteration

- Test both Figma (for full coverage) and code prototype (for realism)
- Iterate based on feedback

Benefits:

- Figma gives full picture (all screens)
- Code prototype validates core interactions (most critical)
- Saves time (not coding everything)

---

14. Deliverables Checklist

14.1 At End of Prototype Phase (Day 26)

Must Have:

☑ Live prototype URL (accessible by anyone)
☑ Source code in GitHub (if code-based)
☑ Dummy data files (all JSON)
☑ User testing report (findings + recommendations)
☑ Validation sign-off (from psikolog, Uwa Farah, parents)
☑ Screenshots of all screens (for documentation)
☑ Handoff document (for developers)

Should Have:

☑ Design system documented (colors, typography, components)
☑ User flow diagrams (visual maps)
☑ Demo video (5-min walkthrough of prototype)
☑ Lessons learned doc (what worked, what didn't)

Nice to Have:

□ Figma file (parallel to code, for designers)
□ Analytics setup (track clicks in prototype - optional)
□ A/B tested variations (if time permits)

14.2 Success Declaration

Prototype Phase is Successful When:

1. ✅ All 5 key user flows are clickable and realistic
2. ✅ 80%+ of testers successfully complete tasks without help
3. ✅ Psikolog team: "Educational approach is sound"
4. ✅ Uwa Farah team: "We can work with this admin panel"
5. ✅ Parent testers: Average NPS > 40 ("Would recommend")
6. ✅ Development team: "Clear what to build, confident we can"
7. ✅ Stakeholders: Unanimous approval to proceed
8. ✅ Timeline met: Completed in 2-3 weeks

If Not All Criteria Met:

- Extend prototype phase by 1 week (iteration)
- OR: Launch development with known risks (document them)
- OR: Pivot concept (if fundamental issues discovered)

---

15. Appendix: Resources & Tools

15.1 Recommended Tools

Design & Mockup:

- Figma (free): figma.com
- Excalidraw (wireframes): excalidraw.com
- Whimsical (user flows): whimsical.com

Front-End Development:

- Vite (React setup): vitejs.dev
- Tailwind CSS: tailwindcss.com
- Headless UI (components): headlessui.com
- Heroicons (icons): heroicons.com
- Framer Motion (animations): framer.com/motion

Dummy Data & Assets:

- Unsplash (photos): unsplash.com
- Pexels (photos): pexels.com
- Lorem Picsum (placeholder images): picsum.photos
- Faker.js (generate fake data): fakerjs.dev
- Mockaroo (CSV generator): mockaroo.com

Hosting & Deployment:

- Vercel (free): vercel.com
- Netlify (free): netlify.com
- GitHub Pages (free): pages.github.com

User Testing:

- Zoom (video calls): zoom.us
- Loom (screen recording): loom.com
- Maze (user testing platform): maze.co
- Google Forms (surveys): forms.google.com

Project Management:

- Notion (docs + tasks): notion.so
- Trello (simple kanban): trello.com
- Miro (whiteboard): miro.com

15.2 Sample Prompts for AI Assistance

For Generating Dummy Data:

```
Generate 10 realistic child development activities for ages 3-5 years in Indonesia.
Each activity should include:
- Title (in Bahasa Indonesia)
- Description (2-3 sentences)
- Materials needed (list)
- 4-5 step instructions
- Educational value (2-3 tags like "Creativity", "Fine Motor")
- Duration estimate
Format as JSON array.
```

For Creating React Components:

```
Create a React component called ActivityCard that displays:
- Activity image (prop: imageUrl)
- Title (prop: title)
- Age range badge (prop: ageMin, ageMax)
- Duration badge (prop: duration)
- Favorite icon (heart, prop: isFavorite)
Use Tailwind CSS for styling. Card should have hover effect and be clickable.
```

For Writing Copy:

```
Write 3 welcome screen slides for a parenting app called JejakAnak:
- Slide 1: Introduce the app (document child's journey)
- Slide 2: Key feature (curated educational activities)
- Slide 3: Value (discover child's talents)
Tone: Warm, supportive, aspirational. Keep each to 2 sentences max.
Target audience: Indonesian parents (use Bahasa Indonesia).
```

15.3 Useful Code Snippets

Basic React Router Setup:

jsx

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Splash from './pages/Splash';
import Welcome from './pages/Welcome';
import Home from './pages/Home';
// ... other pages

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Splash />} />
        <Route path="/welcome" element={<Welcome />} />
        <Route path="/signup" element={<SignUp />} />
        <Route path="/home" element={<Home />} />
        <Route path="/library" element={<ActivityLibrary />} />
        <Route path="/activity/:id" element={<ActivityDetail />} />
        <Route path="/log" element={<LogActivity />} />
        <Route path="/timeline" element={<Timeline />} />
        <Route path="/insights" element={<ChildDashboard />} />
        {/* ... more routes */}
      </Routes>
    </BrowserRouter>
  );
}
```

Loading JSON Data:

```jsx
const [activities, setActivities] = useState([]);
const [loading, setLoading] = useState(true);

useEffect(() => {
  fetch('/data/activities.json')
    .then(res => res.json())
    .then(data => {
      setActivities(data);
      setLoading(false);
    })
    .catch(err => {
      console.error('Failed to load activities:', err);
      setLoading(false);
    });
}, []);

if (loading) return <div>Loading...</div>;
```

Simple Filter Logic:

jsx

```jsx
const [filters, setFilters] = useState({
  category: 'all',
  ageMin: 0,
  ageMax: 180,
  search: ''
});

const filteredActivities = activities.filter(activity => {
  // Category filter
  if (filters.category !== 'all' &&
      !activity.categories.includes(filters.category)) {
    return false;
  }

  // Age filter (child's age within activity range)
  const childAge = selectedChild.ageMonths;
  if (childAge < activity.ageMinMonths ||
      childAge > activity.ageMaxMonths) {
    return false;
  }

  // Search filter
  if (filters.search &&
      !activity.title.toLowerCase().includes(filters.search.toLowerCase())) {
    return false;
  }

  return true;
});
```

Confetti Animation (Simple CSS):

```jsx
// Using react-confetti library
import Confetti from 'react-confetti';
import { useState } from 'react';

function SuccessScreen() {
  const [showConfetti, setShowConfetti] = useState(true);

  useEffect(() => {
    // Stop confetti after 5 seconds
    const timer = setTimeout(() => setShowConfetti(false), 5000);
    return () => clearTimeout(timer);
  }, []);

  return (
    <div>
      {showConfetti && <Confetti />}
      <h1>Activity Logged! 🎉</h1>
      <p>+10 XP earned</p>
    </div>
  );
}
```

---

16. Conclusion

This Dummy Prototype Master Plan provides a comprehensive roadmap for building a high-fidelity, interactive web prototype of JejakAnak in 2-3 weeks. By investing this time upfront, the team will:

1. **Validate Requirements**: Ensure all features are actually needed and well-designed
2. **Test UX**: Identify and fix usability issues before expensive development
3. **Align Stakeholders**: Get everyone on the same page (psikolog, educators, parents, tech team)
4. **Accelerate Development**: Provide clear blueprint for Flutter developers
5. **Reduce Risk**: Catch major problems early when they're cheap to fix

Key Success Factors:

- ✅ Realistic dummy data (not lorem ipsum)
- ✅ Interactive flows (clickable, not just static mockups)
- ✅ Actual user testing (5-10 people, structured sessions)
- ✅ Stakeholder validation (psikolog sign-off critical)
- ✅ Clear handoff (documentation for developers)

Expected Outcomes:

- Validated prototype (approved by all stakeholders)
- Refined requirements (10+ improvements from feedback)
- Confident team (clear vision, ready to build)
- Faster development (fewer surprises, less rework)

Next Steps:

1. Review this plan with team (30 min)
2. Assign prototype builder (1 developer)
3. Start Week 1 on Monday (setup + data prep)
4. Ship prototype by end of Week 3
5. Start development Week 4 with validated design

Remember: The prototype is disposable. Its value is in the learning, not the code. Don't over-engineer it. Build just enough to test hypotheses and validate decisions. Then throw it away and build the real thing properly in Flutter.

Let's build something parents will love! 🚀

Document Version: 1.0
Created: October 20, 2025
Owner: JejakAnak Product Team
Status: Ready for Execution
Timeline: Week 1-3 of Project Roadmap

END OF DUMMY PROTOTYPE MASTER PLAN