

Dokumen Desain & Teknis Super Detail (GDD) - RuangXplore

Judul Proyek: RuangXplore - Penjaga Geometri

Versi: 2.1 (Penambahan Detail Alur Gameplay)

1. Visi & Konsep Inti Proyek

1.1. Pernyataan Visi

"Menciptakan sebuah game petualangan 3D yang mengubah proses belajar Matematika (khususnya bangun ruang) menjadi sebuah pengalaman yang seru, interaktif, dan tak terlupakan bagi siswa Sekolah Dasar, dengan menggabungkan cerita yang menarik, eksplorasi dunia fantasi, dan mekanisme pertarungan kuis yang cerdas."

1.2. Pilar Desain Inti

1. **Edukasi yang Menyenangkan:** Setiap mekanisme gameplay harus memiliki tujuan edukasi yang jelas, namun dibungkus dalam aktivitas yang terasa seperti bermain, bukan belajar.
2. **Petualangan Bercerita:** Progresi pemain didorong oleh narasi dan rasa penasaran untuk mengungkap misteri, bukan hanya menyelesaikan level.
3. **Aksi Cerdas:** "Pertarungan" dimenangkan oleh pengetahuan (menjawab kuis), bukan kecepatan menekan tombol, memberikan rasa pencapaian intelektual.

1.3. Spesifikasi Teknis Umum

- **Game Engine:** Godot Engine 4.2 atau lebih baru.
- **Platform Target:** Android (Low-to-Mid End).
- **API Level Target:** Android API Level 24+ (Nougat 7.0).
- **Orientasi:** Landscape.
- **Gaya Visual:** 3D Low-Poly dengan warna-warna cerah dan kontras tinggi.
- **Renderer:** Vulkan Mobile.

2. Cerita, Karakter, & Dunia Game

2.1. Sinopsis Cerita Lengkap (Lore)

Di sebuah semesta paralel, terdapat dunia ajaib bernama **Mathearia**, sebuah dunia yang seluruhnya terbuat dari konsep matematika dan logika. Suatu hari, sebuah eksperimen energi di Mathearia gagal total, menyebabkan getaran realitas yang membuat dunia tersebut **bertabrakan sesaat** dengan dunia kita.

Tabrakan ini menciptakan fenomena anomali:

- Anak-anak di sekolah pemain tiba-tiba "terhapus" dari realitas, terperangkap di

antara dua dunia.

- Muncul entitas liar bernama "**Monster Kacau**". Monster ini adalah manifestasi fisik dari kebingungan dan konsep yang tidak teratur akibat tabrakan dunia. Mereka tertarik pada area dengan aktivitas intelektual tinggi seperti sekolah.

Tiara, seorang penjaga muda dari Mathearia, berhasil menyeberang ke dunia kita untuk mencari bantuan. Dia menemukan pemain, yang entah kenapa memiliki resonansi kuat dengan energi Mathearia. Menurut Tiara, pemain adalah satu-satunya yang bisa melihat dan menggunakan "**Kekuatan Geometri**"—kemampuan untuk memahami dan memanipulasi ilmu bangun ruang—untuk menstabilkan anomali dan mengalahkan Monster Kacau.

Misi pemain adalah menjadi "**Penjaga Geometri**", berpetualang melalui portal-portal anomali yang muncul di sekolah, mempelajari kekuatan Geometri dari setiap bangun ruang, mengalahkan Monster Kacau, dan pada akhirnya menyelamatkan semua teman yang hilang.

2.2. Profil Karakter & Musuh

- **Pemain (Sang Penjaga):**
 - **Visual:** Karakter anak-anak (laki-laki/perempuan, bisa dipilih) yang memakai seragam sekolah dengan tambahan jaket hoodie modern dan sepatu sneakers. Desainnya simpel dan mudah dianimasikan.
 - **Peran:** Protagonis utama. Tidak banyak bicara, bertindak sebagai perwujudan aksi dari pemain.
- **Tiara (Sang Pemandu):**
 - **Visual:** Gadis remaja dengan rambut panjang berwarna cerah (misal: biru muda). Memakai pakaian yang merupakan gabungan antara jubah akademis dan pakaian petualang fantasi, dengan ornamen berbentuk simbol-simbol geometri.
 - **Peran:** Pemandu, pemberi materi pembelajaran, dan pemberi misi. Kepribadiannya ceria, pintar, dan penyemangat.
- **Monster Kacau:**
 - **Slime Kubus (Level 1):** Monster lendir semi-transparan berbentuk kubus yang goyang-goyang. Mudah dikalahkan, merepresentasikan konsep dasar.
 - **Golem Balok (Level 2):** Monster batu berbentuk balok-balok yang tersusun canggung. Gerakannya lambat tapi terlihat kokoh, merepresentasikan konsep volume dan struktur.
 - **Ular Kristal (Level 3):** Monster berbentuk ular panjang yang tubuhnya terdiri

dari segmen-segmen kristal prisma segitiga yang bersinar. Gerakannya meliuk-liuk, merepresentasikan konsep bentuk yang lebih kompleks.

3. Alur Permainan & Desain Level Super Detail

3.1. Flowchart Umum Permainan

Mulai Game -> MainMenuScene -> Hub_SekolahScene -> (Interaksi dengan Portal Kubus) -> Memuat Level_01_Kubus -> (Menyelesaikan Puzzle & Kuis) -> Kembali ke Hub_SekolahScene -> (Portal Balok Terbuka) -> Memuat Level_02_Balok -> (Dst...) -> Ending "Bersambung..."

3.2. Alur Permainan dari Perspektif Pemain (Gameplay Loop)

Bagian ini menjelaskan pengalaman bermain dari awal hingga menyelesaikan satu siklus level, sesuai dengan permintaan Anda.

1. **Memulai Petualangan di Area Aman:** Pemain memulai permainan di **Hub_SekolahScene**. Ini adalah area aman (*safe point*) di mana tidak ada musuh. Pemain bisa **bebas berjalan-jalan** menggunakan stik analog virtual untuk menjelajahi lobi dan lorong sekolah.
2. **Pertemuan dan Dialog Pertama:** Di tengah lobi, pemain akan melihat karakter **Tiara**. Saat pemain mendekatinya, sebuah ikon interaksi akan muncul. Dengan menekan tombol interaksi, **Sistem Dialog** akan aktif. Sebuah kotak dialog akan muncul di bagian bawah layar, menampilkan potret Tiara dan teks percakapannya yang menjelaskan situasi dan misi awal.
3. **Menuju Portal Pertama:** Setelah dialog selesai, Tiara akan memandu pemain menuju pintu "Kelas Kubus" yang bersinar. Di dalam kelas tersebut, terdapat sebuah portal magis yang berputar-putar. Berinteraksi dengan portal ini akan memicu transisi dan memuat **Level_01_Kubus**.
4. **Menyelesaikan Misi di Level 1:** Di dalam dunia level, pemain akan mengikuti alur yang telah ditentukan: menerima penjelasan materi dari Tiara, memecahkan puzzle (contoh: Drag & Drop Fragmen Kubus), dan membuka jalan ke area selanjutnya.
5. **Pertarungan Kuis Melawan Musuh:** Di area berikutnya, pemain akan melihat **Slime Kubus** yang menghalangi jalan. Tidak ada tombol serangan biasa. Sebagai gantinya, **saat pemain mendekati (encounter) musuh tersebut**, permainan akan otomatis masuk ke mode "**Pertarungan Kuis Instan**".
 - o Layar akan bertransisi, kamera akan fokus pada pemain dan musuh.
 - o Sebuah panel UI kuis akan muncul dengan pertanyaan pilihan ganda tentang bangun ruang Kubus.

- Jika pemain menjawab **benar**, musuh akan langsung kalah dan menghilang.
 - Jika pemain menjawab **salah**, nyawa pemain akan berkurang, dan pemain diberi kesempatan untuk menjawab lagi.
6. **Kembali ke Sekolah (Safe Point):** Setelah musuh dikalahkan dan level selesai, sebuah portal kembali akan muncul. Memasuki portal ini akan mengembalikan pemain ke **Hub_SekolahScene**. Semua progres (level selesai) akan otomatis tersimpan.
7. **Membuka Level Selanjutnya:** Setibanya kembali di sekolah, pemain akan melihat bahwa pintu "Kelas Balok" kini juga bersinar, menandakan bahwa level berikutnya telah terbuka. Siklus permainan (masuk portal -> belajar -> puzzle -> kuis battle -> kembali) pun berlanjut.

3.3. Rincian Desain Setiap Scene

- **MainMenuScene.tscn**
 - **Visual:** Latar belakang berupa pemandangan sekolah yang sepi dengan efek cahaya misterius.
 - **UI Nodes:** VBoxContainer berisi Button "Mulai Petualangan", Button "Pengaturan", Button "Keluar".
- **Hub_SekolahScene.tscn**
 - **Visual:** Lingkungan 3D berupa lobi sekolah dan satu lorong. Terdapat 3 pintu kelas. Awalnya, hanya pintu "Kelas Kubus" yang bersinar.
 - **Objek Interaktif:** Tiara, Papan Tulis Ajaib (Save Point), Portal di setiap kelas.
- **Level_01_Kubus.tscn**
 - **Desain Lingkungan:** Dunia fantasi yang terdiri dari pulau-pulau rumput melayang berbentuk kubus. Palet warna cerah: hijau, biru langit, putih awan.
 - **Alur Level:** Dialog pembelajaran -> Puzzle Drag & Drop -> Encounter Slime Kubus -> Kembali.
- **Level_02_Balok.tscn**
 - **Desain Lingkungan:** Reruntuhan desa kuno di tengah hutan. Bangunan, tembok, dan jalan terbuat dari balok batu dan kayu. Palet warna bumi: coklat, abu-abu, hijau lumut.
 - **Alur Level:** Dialog pembelajaran -> Puzzle Susun Balok -> Encounter Golem Balok -> Kembali.
- **Level_03_Prisma.tscn**
 - **Desain Lingkungan:** Gua kristal bawah tanah yang gelap namun diterangi oleh kristal-kristal prisma segitiga yang bersinar. Palet warna magis: ungu, biru tua, pink terang.
 - **Alur Level:** Dialog pembelajaran -> Puzzle Pantul Cahaya -> Encounter Ular

Kristal -> Kembali.

4. Spesifikasi Teknis Super Detail (Godot 4)

4.1. Sistem Pemain (Player.tscn)

- **Struktur Node:** CharacterBody3D (induk, di grup "Player") dengan node anak CollisionShape3D, MeshInstance3D, AnimationPlayer, Camera3D, dan RayCast3D.
- **Skrip player.gd:** Mengelola input dari Input.get_vector() untuk gerakan via move_and_slide() di _physics_process. RayCast3D digunakan untuk deteksi interaksi.

4.2. Sistem Interaksi (Signals & Groups)

- Gunakan grup **Interactable**. Skrip dasar interactable.gd berisi signal interacted dan fungsi interact() untuk di-override oleh NPC, puzzle, dll.

4.3. Sistem Dialog (DialogueManager.gd - Autoload/Singleton)

- Gunakan **Resource DialogueResource.tres** (class_name DialogueResource) untuk menyimpan data dialog (nama, potret, kalimat).
- DialogueManager akan memunculkan DialogueUI.tscn dan mengisi datanya, menggunakan Tween atau Timer untuk efek ketik.

4.4. Sistem Kuis Battle (QuizBattleManager.gd - Autoload/Singleton)

- Musuh (Enemy.tscn) memiliki Area3D yang saat sinyal body_entered terpanggil, akan memicu QuizBattleManager.start_battle(self, question_bank).
- QuizBattleManager akan meng-pause game (get_tree().paused = true), memunculkan QuizBattleUI.tscn, dan mengelola logika kuis. Jika benar, panggil enemy.queue_free(). Jika salah, kurangi nyawa pemain.

4.5. Sistem Penyimpanan (SaveManager.gd - Autoload/Singleton)

- Gunakan Dictionary untuk menyimpan data progres (cth: {"level_unlocked": 1}).
- Simpan ke file user://savegame.json menggunakan FileAccess dan JSON.stringify().

5. Daftar Aset Super Detail

Tabel Aset 3D (Low Poly)

Nama Aset	Estimasi Polycount	Deskripsi Visual
Player_Character	< 1500 tris	Model karakter utama dengan rig untuk animasi dasar.

Tiara_NPC	< 1500 tris	Model Tiara dengan rig untuk animasi dasar.
Slime_Kubus	< 500 tris	Kubus semi-transparan dengan mata, rig sederhana untuk goyang.
Golem_Balok	< 1000 tris	Kumpulan balok batu yang membentuk humanoid, rig kaku.
Ular_Kristal	< 1200 tris	Serangkaian prisma yang terhubung, rig tulang (bone) untuk meliuk.
Lingkungan		
Sekolah_Hub	< 5000 tris	Lobi, lorong, 3 pintu. Tekstur minimalis.
Dunia_Kubus_Set	< 8000 tris	Pulau, jembatan, pohon kubus, altar mesin, fragmen kubus.
Desa_Balok_Set	< 8000 tris	Rumah balok, tembok, gerbang, kotak kargo, jalan batu.
Gua_Prisma_Set	< 8000 tris	Dinding gua, stalaktit/stalagmit, kristal prisma berotasi.

Tabel Aset 2D & UI

Nama Aset	Ukuran (px)	Deskripsi Visual
Potret_Player	256x256	Ekspresi netral karakter pemain.
Potret_Tiara	256x256	Ekspresi tersenyum karakter Tiara.
UI_Tombol_Utama	~300x100	Tombol dengan style

		kayu/batu fantasi.
UI_Panel_Dialog	~1800x400	Panel transparan gelap di bawah layar.
UI_Panel_Kuis	~1600x800	Panel di tengah layar untuk pertanyaan dan 4 tombol jawaban.
lkon_Hati	128x128	lkon hati merah untuk menunjukkan nyawa.

Tabel Aset Audio

Nama Aset	Tipe	Deskripsi Audio
BGM_Sekolah	Musik	Musik piano yang tenang, sedikit melankolis dan misterius.
BGM_DuniaKubus	Musik	Musik orkestra yang ceria, penuh semangat petualangan.
BGM_KuisBattle	Musik	Musik elektronik cepat dengan beat menegangkan.
SFX_Klik_UI	Efek Suara	Suara "pop" atau "klik" yang memuaskan.
SFX_Jawaban_Benar	Efek Suara	Suara "ding!" atau jingle pendek yang ceria.
SFX_Jawaban_Salah	Efek Suara	Suara "buzz" atau "thud" yang rendah.
SFX_Musuh_Hilang	Efek Suara	Suara "poof" seperti kepulan asap.
SFX_Portal_Aktif	Efek Suara	Suara "humming" atau dengungan magis yang konstan.

6. Strategi Optimasi Performa (Android Low-End)

1. **Rendering:** Gunakan Occlusion Culling dan MultiMeshInstance3D untuk objek berulang guna menjaga Draw Calls tetap rendah.
2. **Model 3D:** Patuhi polycount yang tertera di tabel aset. Gunakan kembali model di mana pun memungkinkan.
3. **Tekstur & Material:** Gunakan **satu palet tekstur (Texture Atlas)** berukuran kecil (256x256 atau 512x512) per level. Kompresi tekstur ke format **ETC2** saat ekspor.
4. **Coding (GDScript):** Gunakan `_physics_process` hanya untuk fisika. Gunakan Timer atau Signal untuk menjalankan kode hanya saat diperlukan daripada di `_process`. Manfaatkan profiler bawaan Godot untuk mencari bottleneck.