Prosiding
Seminar Nasional
Unit Kegiatan Mahasiswa Penalaran dan Riset
IKIP PGRI Bojonegoro
Tema "Eksplorasi Penalaran dalam Riset untuk Meningkatkan Kualitas Publikasi Ilmiah"

Pengembangan Keterampilan Pemilihan Kata secara Interaktif melalui Permainan Petualangan Edukasi Menangkap Kata berbasis Scratch

Dimas Rizqia Hendinata[1](✉), Martinus Seriandika Saputra[2], Cahyo Hassanudin[3]
Pendidikan Bahasa dan Sastra Indonesia, IKIP PGRI Bojonegoro, Indonesia
dimashendinata21@gmail.com

abstrak— Scratch adalah bahasa pemrograman untuk pembelajaran interaktif. Interaktif merujuk pada hubungan timbal balik yang saling berkesinambungan antarindividu. Pilihan kata ialah kemampuan membedakan makna secara tepat, memilih kata sesuai situasi, serta mempertimbangkan nilai rasa dalam masyarakat. Penelitian ini mengadopsi pendekatan model SDLC-waterfall yang mencakup lima tahap utama, yaitu: pengumpulan requirement, perancangan design, pelaksanaan implementation, pengujian dan verifikasi verification, serta pemeliharaan maintenance. Setiap fase saling berkesinambungan dan memastikan pengembangan sistem yang sistematis dan terstruktur Penemuan ini menghasilkan sprite yang dipakai dalam pembuatan game yaitu 1) Awal Permainan 2) Kata (benda, kerja, sifat) 3) Sepeda motor 4) Mobil 5) Kata benda pada lingkaran kuning 6) Pembatas jalan 7) Mobil Van 8) Palang 9) Bis sekolah 10) Kata kerja pada lingkaran kuning 11) Pesawat luar angkasa 12) Meteor 13) Planet 14) Bintang 15)Background pada scratch.
Kata kunci— Scratch, Interaktif, Pemilihan kata

Abstract—Scratch is a programming language for interactive learning. Interactive refers to a continuous, reciprocal relationship between individuals. Word choice is the ability to distinguish meaning appropriately, choose words according to the situation, and consider the value of taste in society. This research adopts the SDLC-waterfall model approach which includes five main stages, namely: requirement gathering, design, implementation, testing and verification, and maintenance. Each phase is interconnected and ensures systematic and structured system development.This invention produces sprites used in making games, namely 1) The beginning of the game 2) Word (noun, verb, adjective) 3) Motorbike 4) Car 5) Noun in yellow circle 6) Roadblock 7) Van 8) Crossbar 9) School bus 10) Verb in yellow circle 11) Spaceship 12) Meteor 13) Planet 14) Star 15) Background on scratch.
Keywords— Scratch, Interactive, Word selection

PENDAHULUAN

Pilihan kata ialah kemampuan untuk membedakan dengan tepat nuansa makna daru sebuah gagasan yang dapat ingin disampaikan, kemampuan menemukan bentuk dari kata yang tepat dengan situasi (Al Ayubi & Sulistyowati, 2023) dan nilai dari rasa yang dimiliki sebuah kelompok masyarakat pemirsa (Djafar, 2020). Pilihan kata ialah kemampuan untuk membedakan dengan tepat nuansa makna daru sebuah gagasan yang dapat ingin disampaikan,

kemampuan menemukan bentuk dari kata yang tepat dengan situasi (Al Ayubi & Sulistyowati, 2023).

Jenis pilihan kata atau bisa disebut juga diksi bisa digolongkan kedalam dua kategori yaitu diksi berdasarkan dari maknanya ialah denotatif makna dari dalam wajar alam secara eksplisit, yang bermakna sesuai apa adanya, yang artinya suatu pengertian sebuah kata yang dikandung secara objektif, sedangkan makna dari konotatif ialah sebuah kata yang dapat berbeda satu kelompok dengan kelompok dari masyarakat lain juga, yang artinya sesuai dengan gaya pandangan hidup atau norma penilaian dari kelompok masyarakat itu (Hardianto & Sucipto, 2017). Yang selanjutnya ada pemilihan kata leksikal yang merupakan perbendaharaan dari kata yang dipunyai oleh bahasa (Suparman, 2019). Leksikal juga salah satu jenis makna di bidang semantik dari dalam bahasa Melayu (Salleh dkk, 2020).

Fungsi diksi sendiri antara lain untuk membuat pendengar dan pembaca tidak mempunyai salah paham dan mengerti dengan benar terhadap apa yang akan disampaikan oleh penulis atau pembicara (Suryaningsih, 2017), untuk menyajikan gagasan kepada orang lain (Tresnawati, 2023) Agar merasakan yang dirasakan oleh pengarang dan tidak terjadinya salah tafsir (Yulistiana, 2019).

Pengertian interaktif ialah suatu hal terkait dengan suatu yang bersifat, saling interaktif dan saling memiliki timbal balik serta saling berhubungan antara seseorang dengan seseorang lainya (Shalikhah, 2016) dalam (Hariadi dkk, 2023). Interaktif juga didasarkan pada sebuah pemikiran bahwa dari media presentasi didasari pada umumnya tidak dilengkapi oleh alat untuk mengarahkan yang disebabkan oleh user (Supardi, 2013). Sedangkan dari pengertian interaktif sendiri terkait dengan komunikasi dari dua arah atau saling mengerjakan aksi bisa dari komponen-komponen komunikasi (Arsyad, 2018).

Orang yang memiliki keinginan untuk belajar lebih tinggi akan melakukan upaya agar belajar lebih efektif dan serius penuh semangat, hal ini begitu juga sebaliknya orang yang kurang keinginan motivasi untuk belajar akan melakukan sebuah kegiatan pembelajaran secara sembrono atau bahkan tidak mau andil untuk berpartisipasi (Aminingtyas & Wardhani, 2023). Artinya disini terjadi hubungan yang positif antara motivasi belajar sama dengan hasil dari belajar, makin naik motivasi diri belajar seseorang, maka semakin naik pula hasil pembelajarannya (Jannah dkk, 2021). Secara umum juga hubungan interaktif yang terkait dalam dunia perbisnisan biasanya terlibat antara ketiga pihak yang saling berhubungan erat dengan yang lainnya (Winarko, 2011).

Manfaat dari adanya interaktif ini ialah siswa memiliki keinginan motivasi belajar yang tinggi di tunjukan dengan semangat siswa untuk mengerjakan tugasnya, perhatian siswa untuk mengikuti sebuah pelajaran, tanggung jawab dari siswa untuk mengerjakan tugasnya, rasa senang dan puas dalam mengerjakan sebuah tugas yang diberi guru dan reaksi positif yang diekspresikan siswa terhadap dorongan yang dilakukan guru (Hidayah dkk, 2017). Proses dari pembelajaran juga akan lebih bermakna dan menyenangkan, sehingga mempengaruhi peningkatan prestasi dari belajar siswa (Shalikhah, 2017). Sejumlah penelitian juga sebelumnya menunjukkan penggunaan dari adanya multimedia interaktif juga mampu meningkatkan prestasi belajar, penguasaan konsep, dan berkemampuan untuk berpikir kritis (Harsiwi & Arini, 2020).

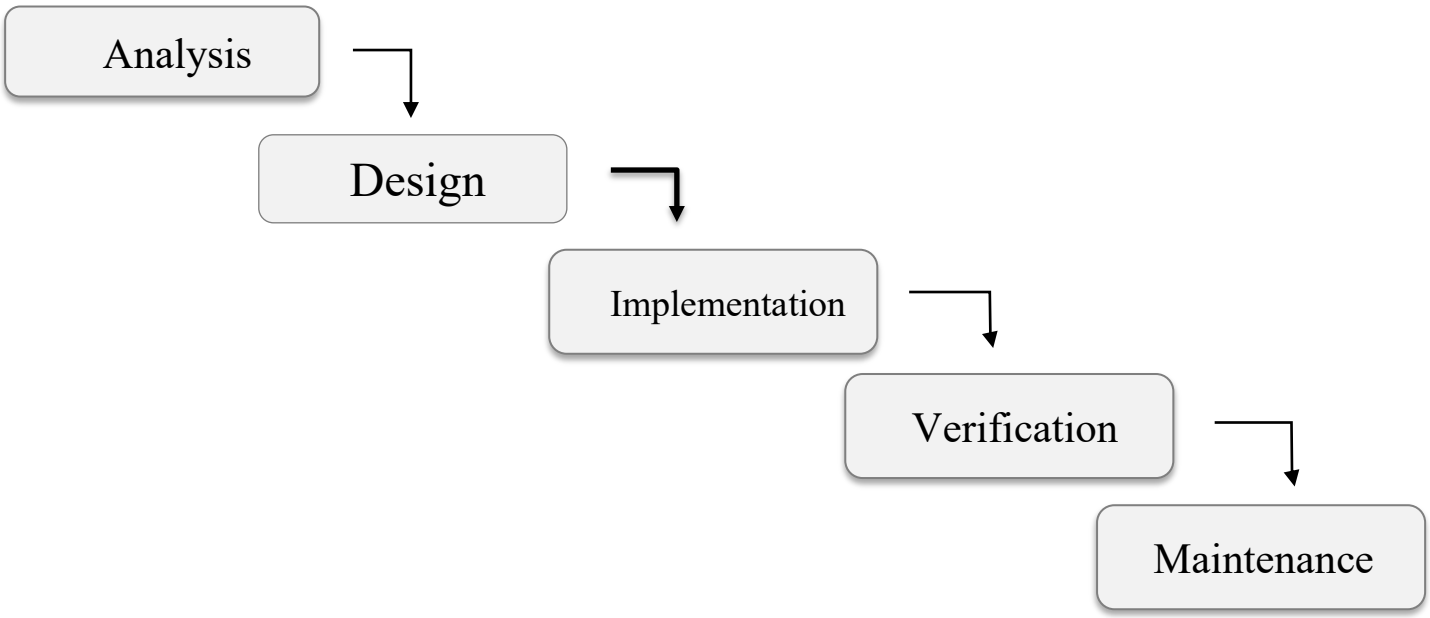
Scratch ialah bahasa dari pemrograman yang digunakan sebagai alat media pembelajaran ke dalam bentuk kuis, animasi, permainan dan lainya (Chaerunnisa & Bernard, 2021). Scratch sendiri merupakan sebuah bentuk aplikasi untuk menciptakan produk tanpa berpikir keras terkait dengan bahasa pemrograman (Sudihartinih & Rachmatin, 2021). MIT MEDIA LAB adalah yang mengembangkan program dari scratch yang dapat dipakai dalam pembelajaran dari anak usia 8 tahun sampai 16 tahun (Zahid, 2021) dalam (Pratiwi & Bernard, 2021).

Penerapan cara penggunaan scratch dapat meningkatkan dari hasil belajar siswa (Hakim & Kasiyun, 2024). Fungsi peran guru pada konteks ini berpengaruh memainkan peran yang sangat penting untuk memotivasi pelajar untuk belajar dengan sedemikian mungkin dengan melalui penerapan dari kurikulum yang menyatukan berbagai strategi mengajar dan memanfaatkan sebuah teknologi secara efisien dan cerdas (Elsola dkk, 2023). Pengembangan dan pembuatan media belajar dengan menggunakan animasi dari dua dimensi berdasarkan pada java scratch bersamaan dengan mengetahui gambaran kisi-kisi efektivitas dan kemampuan media pembelajaran animasi dari dua dimensi java scratch dengan pemanfaatan sebagai sebuah alat pembelajaran memahami sifat dari bangun ruang, pembelajaran fisika teori dari kinetik gas di SMA dan masih banyak lainnya (Martanti dkk, 2013).

Pelatihan coding dengan berbasis scratch sudah terbukti menaikkan pemahaman terhadap peserta didik dengan konsep-konsep dari dasar pemrograman di berbagai kalangan negara (Najuah, 2022) dalam (Qurin, 2024). Selain itu bagi siswa lebih tepatnya siswa yang berkebutuhan khusus dalam menangani kesulitan di kelas inklusi di dalam pemahaman perkalian bilangan, menaikkan kemampuan visual, konsentrasi belajar siswa, berhitung dan membaca bagi pelajar terutama pelajar berkebutuhan khusus pada kelas inklusi (Jannah dkk, 2021). Keunggulan lainya dari scratch juga melatih kreativitas pelajar dan memberikan kesiapan pada siswa untuk menambah pemahaman pemrograman (Hardiansyah dkk, 2023).

METODE PENELITIAN

Penelitian ini menggunakan metode SDLC (Sistem Development Life Cycle) waterfall model. SDLC merupakan pendekatan yang mempunyai tahapan untuk membangun dan menganalisa suatu sistem rancangan dengan menggunakan sebuah siklus yang spesifik terhadap aktivitas kegiatan pengguna contohnya metode waterfall (Munthe, 2017). Metode waterfall ialah metode pengembangan suatu perangkat lunak pembuatan sistem yang memungkinkan dilakukan secara sistematis dan terstruktur sesuai dengan priode/siklus pengembangan yang sudah ada (Bashari, 2024). Ada 5 tahapan dalam metode ini seperti gambar dibawah ini:

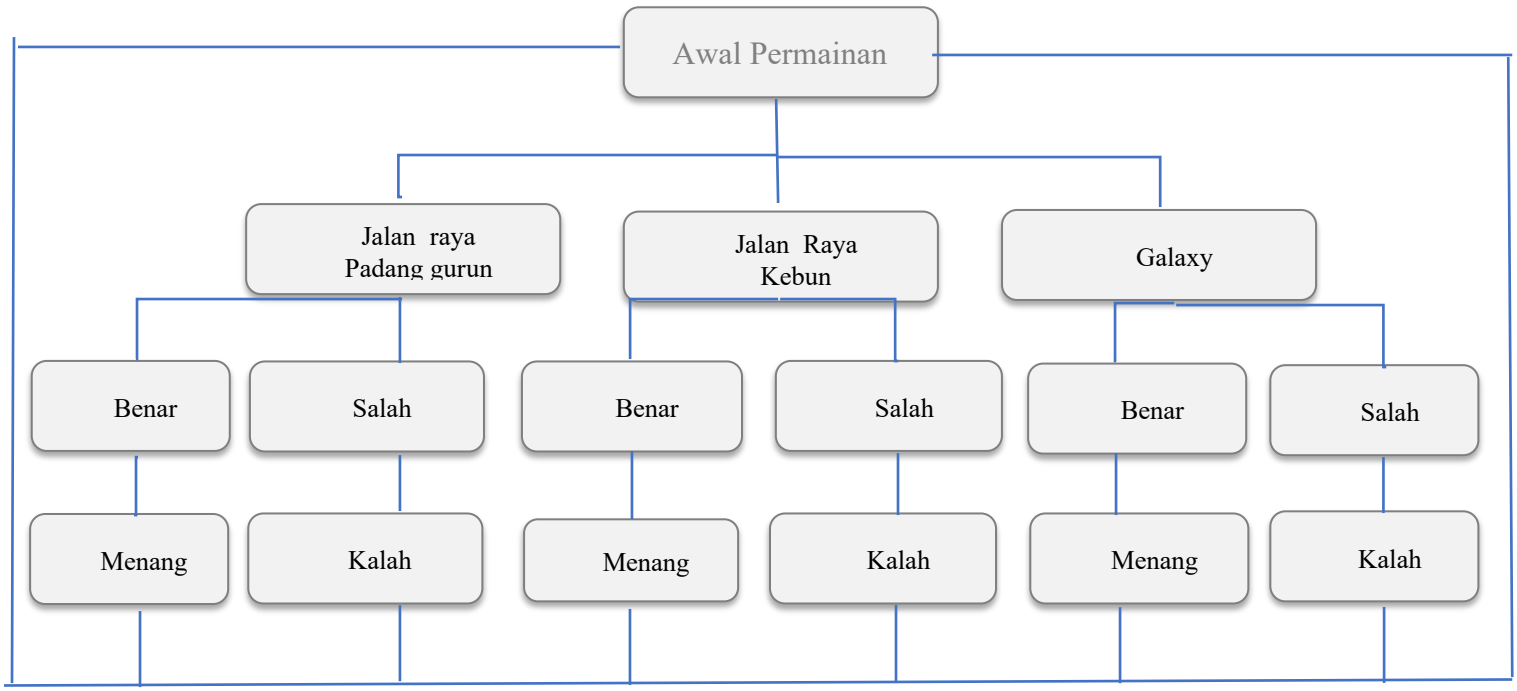


1. *Requirements*

Pada tahapan ini, definisi dari kebutuhan konsumen atau dari para pengguna yang terkait dengan transformasi sebuah sistem kerja untuk dijadikan perangkat lunak software (Nasrullah, dkk, 2021). Ambiguitas yang paling sering muncul dari software requirements ialah jenis ambiguitas leksikal atau bisa disebut lexical ambiguity (Mukhlis, 2022). Requirements yang ada dari sistem stakeholder kemudian dianalis tim QA agar mengetahui inti detail dari software, fitur, module dan fungsi dari yang akan di share, melakukan validasi dan review jika terdapat kekurangan dan untuk melengkapi agar lebih memperjelas requirements tersebut, menganalisa sesuatu yang bisa di uji secara automated dan manual serta menganalisa dari cakupan fungsi/fitur apa saja yang akan diperiksa dan diuji secara non-functional dan functional (Dhaifullah, 2022).

2. *Design*

Desain perancangan program dari perangkat lunak keseluruhan termasuk, seni bangun perangkat lunak, struktur data, representasi antarmuka atau prosedur dari pengkodean (Badrul, 2021). Pada tahapan ini, melakukan desain sistem dengan memakai permodelan Unified Modelling Language (UML) dan pada tahapan ini melakukan perancangan antar muka website ataupun aplikasi yang akan dibikin (Fachri & Surbakti, 2021).



3. *Implementation*

Pada tahapan ini, mulai mengubah ide menjadi sebuah kenyataan dan bisa digunakan (Ardiansyah, 2020). Plan implementation juga bisa disebut pelaksanaan rencana agar mewujudkan karangan rencana yang ditulis kedalam penjabaran rencana dalam perbuatan ilmiah untuk menentukan apakah rencana itu efektif dan baik (Albab, 2021).

4. *Verfication*

Tahapan selanjutnya ialah verification, pada tahapan ini seluruh unit yang diupgrade atau dikembangkan pada tahapan implementasi digabungkan kedalam sistem sesudah dilakukannya pengujian pada seluruh unit (Pratiwi dkk, 2023). Setelah itu dilakukan

pengujian agar bertujuan mengetahui software sudah pas swsuai desain dan apakah masih terdapat ada kesalahan (Sitanggang dkk, 2023).

5. *Maintenance*

Pada tahapan ini, sistem diberikan kepada pemakai untuk dipakai sesuai hak aksesnya sesudah tahap pengujian program dan revisi sesuai yang diperlukan (Wijaya dkk, 2023). Perangkat lunak butuh perawatan dalam sebuah proses pengembangan dikarenakan bisa saja kejadian eror kecil maupun yang besar (Yusron & Huda, 2021).

HASIL DAN PEMBAHASAN

Scratch adalah media pemrograman visual berbasis blok yang dikembangkan oleh MIT Media Lab untuk mendukung pembelajaran dasar pemrograman dan pengembangan kreativitas, khususnya bagi pelajar dan pemula. Dengan Scratch, pengguna dapat menciptakan proyek interaktif seperti animasi, simulasi, dan permainan tanpa harus menulis kode dalam format teks yang rumit.

1. Awal Permainan

Gambar 1. Pemandu Permainan https://images.app.goo.gl/fdND2Q8EdkwFVZVJ8. https://images.app.goo.gl/NPP8PFBbV61fiwoy9, https://images.app.goo.gl/PtrZ9QSrrUXfcYZ46. https://images.app.goo.gl/tsXdKbZh2opRjojY7.

langkah-langkah untuk menggerakkan karakter dan mengatur interaksi di latar Awal Permainan. Pertama, pergi ke tab *Events* dan pilih blok *when (green flag) clicked*. Lalu, pergi ke tab *Looks*, pilih *hide* dan *switch backdrop to (Awal permainan)* untuk mengganti latar ke *Awal Permainan*. Selanjutnya, pergi ke tab *Control*, tarik blok *Forever*, dan tambahkan blok *If (condition) then* ke dalamnya. Dalam tab Motion, pilih *key (Right Arrow) pressed?*, masukkan ke dalam kondisi *If (condition*), kemudian tambahkan *point direction (90), move (5) steps*, dan *next costume* ke dalam blok *If (condition)*. Setelah ini, duplikasikan blok *If (condition),* letakkan di bawahnya, lalu tambahkan blok *delete this clone* dari tab *Control*. Langkah ini digunakan untuk menggerakkan karakter dalam latar *Awal Permainan*. Coding kedua, yang digunakan untuk membuat karakter berbicara dan menghilang saat tombol di samping kanan ditekan, lakukan langkah-langkah berikut. Pertama, pergi ke tab *Events* dan pilih *when (green flag) clicked*. Kemudian, pergi ke tab *Looks*, pilih *say (Hello!) for (0) seconds* sebanyak empat kali, isi dengan teks yang ingin disampaikan di awal permainan, serta atur durasi pesan. Tambahkan blok *hide* di bawahnya untuk menyembunyikan sprite

setelah pesan selesai ditampilkan. Terakhir, untuk menghilangkan sprite berdasarkan perubahan latar, tambahkan blok *when backdrop switches to (Latar tiga game di samping)*, kemudian tambahkan blok *hide* di bawahnya. Duplikasikan blok ini sebanyak dua kali jika diperlukan. Dengan langkah-langkah ini, program akan berjalan sesuai desain interaktif yang diinginkan.

2. Kata (benda, kerja, sifat)

Gambar 2. https://images.app.goo.gl/NPP8PFBbV61fiwoy9. https://images.app.goo.gl/PtrZ9QSrrUXfcYZ46.

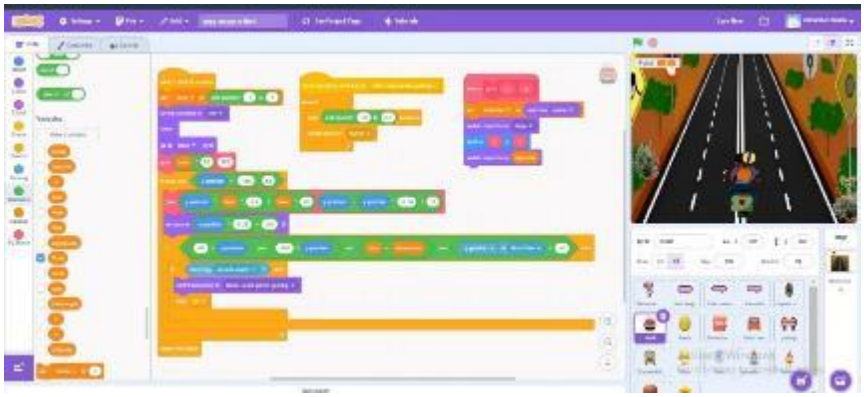
Pada gambar 2 dalam pengembangan proyek Scratch, berpindah latar (backdrop) dan mengatur tampilan objek (sprite) merupakan langkah penting. Langkah pertama adalah memastikan objek yang relevan terlihat saat program dijalankan: pergi ke tab *Events*, kemudian tarik blok *when (green flag)* clicked ke area coding. Selanjutnya, pergi ke tab *Looks* dan tarik blok *show*. Hubungkan blok ini dengan blok *when (green flag) clicked* untuk memastikan objek terlihat di awal program. Untuk mengatur perpindahan latar ketika suatu sprite diklik, pergi ke tab *Events* dan tarik blok *when this sprite clicked.* Pergi ke tab *Variables*, klik *Make a Variable*, lalu buat beberapa variabel sesuai kebutuhan, seperti benda, kerja, dan sifat, kemudian klik OK. Kembali ke tab *Looks,* tarik blok *switch backdrop to (nama latar)* dan ganti *(nama latar)* dengan nama latar yang sudah disiapkan. Pergi ke tab *Events*, tarik *blok broadcast (pesan)*, dan buat pesan baru seperti kata kerja untuk menentukan aksi tertentu. Akhiri langkah ini dengan menambahkan blok *hide* dari tab *Looks* untuk menyembunyikan sprite setelah fungsinya selesai dijalankan. Untuk menyembunyikan soal yang tidak diklik, pergi ke tab *Events*, tarik *blok when I receive (pesan),* pilih pesan yang sesuai seperti kata benda atau kata sifat, lalu pergi ke tab *Looks* dan tarik blok *hide*, kemudian hubungkan dengan blok *when I receive (pesan).* Contoh alur program: ketika *green flag* diklik, semua sprite yang relevan akan terlihat, ketika sprite tertentu diklik, latar akan berpindah sesuai dengan logika yang sudah ditentukan, variabel akan digunakan untuk menyimpan nilai tertentu, dan sprite yang tidak relevan akan disembunyikan. Pastikan nama latar dan pesan yang digunakan konsisten agar tidak terjadi konflik dalam logika program. Gunakan variabel untuk mempermudah pengelolaan data dan meningkatkan fleksibilitas program. Langkah-langkah ini memastikan program Scratch berjalan sesuai dengan desain interaktif yang diinginkan. Dengan struktur yang sistematis, kode menjadi lebih mudah dipahami dan dikelola.

3. Sepeda motor

Gambar 3. pngtree-a-motorcyclist-couples-rear-view-male-white-rider-vector-picture-image_10302824. https://images.app.goo.gl/PfRtuzzafbcgeKaZ9.

Dalam sebuah permainan interaktif, terdapat serangkaian langkah pengkodean yang dirancang untuk mengelola elemen-elemen visual dan fungsional seperti gerakan sepeda motor, suara, dan tampilan latar. Untuk memastikan sepeda motor tidak langsung terlihat saat permainan dimulai, digunakan blok *When (greenflag) clicked* diikuti oleh perintah *hide* pada bagian *Looks*. Hal ini memastikan bahwa sepeda motor hanya muncul pada momen tertentu. Efek suara diintegrasikan dengan cara menambahkan blok *When I start as a clone* diikuti oleh loop *Forever*. Dalam loop ini, blok *start sound (Car passing)* digunakan untuk menciptakan efek suara saat sepeda motor muncul, memberikan kesan realistis. Untuk menampilkan sepeda motor berdasarkan latar tertentu, digunakan blok *When backdrop switches to* pada bagian *Event*, yang dipadukan dengan perintah *Show* pada *Looks*. Blok ini diduplikasi untuk menangani dua kondisi, yaitu ketika latar berubah menjadi *Latar Jalan Raya Padang Pasir* dan *Kamu Menang.* Gerakan horizontal diatur dengan membuat *variabel* bernama *PlayerLane*. Blok *When backdrop switches to (Latar Jalan Raya Padang Gunung)* digunakan sebagai pemicu. Dalam loop *Forever*, operator *(condition) And (condition* digunakan untuk menentukan kondisi pergerakan tombol panah kiri dan kanan. Jika tombol panah kiri ditekan, *variabel PlayerLane* berubah sebesar *-1*, sedangkan tombol panah kanan mengubahnya sebesar *1*. Untuk memastikan respons yang lebih halus, ditambahkan blok *Wait until* dengan operator *(condition) Or (condition)* yang memeriksa apakah tombol panah kiri atau kanan ditekan sebelum perubahan nilai variabel terjadi. Untuk gerakan melompat, blok *When backdrop switches to (Latar Jalan Raya Padang Gurun)* dipadukan dengan loop *Forever*. Dalam loop tersebut, kondisi *If (Key up arrow pressed)* digunakan bersama operator *(condition) And (condition)* untuk memeriksa posisi vertikal *(y position)* dengan nilai *-100*. *Variabel yVelocity* diatur ke nilai *40* untuk memulai lompatan, lalu secara bertahap dikurangi dengan nilai -6 untuk menciptakan efek gravitasi. Jika posisi y lebih kecil dari -100, variabel yVelocity diatur ke 0, dan posisi y dikembalikan ke -100, menjaga batas bawah gerakan. Kostum sepeda motor diatur berdasarkan nilai variabel *PlayerLane*. Operator Join digunakan untuk menggabungkan nama kostum sesuai jalur yang dipilih pemain, memastikan tampilan visual yang sesuai. Dengan pendekatan ini, pengembang dapat menciptakan pengalaman bermain yang dinamis dan imersif. Kombinasi logika kondisi, variabel, dan manipulasi visual memastikan permainan berjalan sesuai dengan skenario yang dirancang.

4. Mobil



Gambar 4. https://images.app.goo.gl/PfRtuzzafbcgeKaZ9. https://images.app.goo.gl/r2vYW4YJJQ6A3Gkb7.

Pada gambar 4. Permainan ini dirancang dengan tiga logika utama yang saling terintegrasi untuk mengatur kloning objek, pergerakan kendaraan, dan modularisasi fungsi objek. Pada bagian pertama, kloning objek kendaraan dilakukan dengan memanfaatkan blok *when I start as a clone* yang mengatur *variabel lane*, menentukan jalur kendaraan secara acak dengan nilai -1, 0, atau 1, dan memposisikan kendaraan pada koordinat awal dengan rumus *x = lane * 70 dan y = 180*. Kostum sprite diatur ke *low*, dan kendaraan ditampilkan menggunakan blok *show*. Kendaraan kemudian bergerak vertikal ke bawah dengan pengurangan posisi *y sebesar -0,2* dalam loop *repeat until*, sementara ukurannya disesuaikan secara dinamis dengan rumus *size = 10 - (y position * 0,04)* untuk menciptakan ilusi perspektif. Jika kendaraan menyentuh objek sepeda motor, latar belakang berubah menjadi *Kamu kalah padang gurung* dan seluruh skrip dihentikan. Apabila kendaraan melewati batas layar dengan kondisi *y position < -180*, klon dihapus menggunakan blok *delete this clone*. Bagian kedua mengatur pemunculan klon kendaraan secara berkala selama latar aktif menggunakan blok *when backdrop switches to* dan *forever*, dengan waktu tunggu acak antara 1 hingga 2 detik melalui rumus *wait (pick random 1.0 to 2.0)* sebelum membuat klon baru menggunakan blok *create clone of myself*. Untuk mendukung efisiensi manipulasi sprite, bagian ketiga menggunakan blok khusus *define goto (x) (y) (costume name),* yang pertama-tama mengubah kostum sprite ke huge, memindahkannya ke koordinat *(x, y)*, lalu mengembalikannya ke kostum asli yang diatur oleh parameter costume name. Ketiga logika ini membentuk dasar permainan yang interaktif, dengan integrasi dinamis antara pengaturan visual, pergerakan objek, dan kondisi permainan.

5. Kata benda pada lingkaran kuning

Gambar 5. https://images.app.goo.gl/Kib3mdXDeCiEkqWa9. https://images.app.goo.gl/PfRtuzzafbcgeKaZ9. https://images.app.goo.gl/LxFQERaTZUTr2iq49. https://images.app.goo.gl/r2vYW4YJJQ6A3Gkb7.

Gambar di atas menampilkan Scratch yang terdiri dari tiga bagian utama, yang bekerja secara berurutan untuk mengatur logika permainan interaktif pertama membuat klon dengan pengaturan acak langkah pertama Pergi ke *Event*, tarik *When I start as a clone.* Pergi ke *Variables,* pilih *variabel lane* dan atur menggunakan *set lane to pick random (1) to (6).* Ini menentukan jalur acak di mana klon akan muncul. Dari *Looks*, tambahkan *switch costume to pick random (1) to (15)* untuk memilih kostum sprite secara acak. Masukkan *show* agar klon terlihat di layar, dan tambahkan *go to front layer* untuk memastikan klon berada di lapisan terdepan. Bagian ini bertanggung jawab untuk menciptakan klon dengan atribut acak seperti jalur dan kostum, serta memastikan klon terlihat di layar. Kedua mengatur pergerakan sprite *Control,* masukkan *repeat until (y position < -180).* Ini membuat sprite terus bergerak ke bawah hingga keluar layar. Dari *Motion*, tambahkan *change y by (-10)* agar sprite bergerak secara vertikal ke bawah. Lalu *Looks*, tambahkan *set size to (y position * -0.22 + 100)%* untuk menyesuaikan ukuran sprite berdasarkan posisinya, sehingga menciptakan efek perspektif. Tambahkan *hide* dari *Looks* untuk menyembunyikan sprite setelah keluar dari layar. Bagian ini memastikan sprite bergerak secara dinamis di layar dengan ukuran yang menyesuaikan, memberikan efek realisme dan rumus yang ketiga deteksi tabrakan dan sistem poin pilih *Control*, masukkan *if (y position < -95 AND y position > -105 AND lane = playerLane)* untuk mendeteksi apakah sprite bersinggungan dengan pemain (sepeda motor). Jika kondisi ini terpenuhi, tambahkan *stop all* dari *Control* untuk menghentikan permainan dari *Sound*, tambahkan *play sound (Coin) until done* untuk memberikan efek suara ketika pemain menghindari sprite. *Variables,* masukkan *change Point by (1)* untuk menambahkan poin jika sprite berhasil dihindari. Bagian ini menangani logika interaksi antara sprite dan pemain, memberikan poin, atau menghentikan permainan jika terjadi tabrakan. Kondisi menang dan hapus klon mulai dari control, tambahkan *if (Point = 35)* untuk mendeteksi apakah poin mencapai angka kemenangan. Jika kondisi ini terpenuhi, tambahkan *switch backdrop to (Kamu menang, padan gunung)* dari *Looks* untuk mengubah latar belakang menjadi tampilan kemenangan. Tambahkan *stop all* untuk menghentikan permainan setelah pemain menang. Terakhir, tambahkan *delete this clone* dari *Control* untuk menghapus klon setelah selesai. Bagian ini menentukan kondisi kemenangan dan menghapus klon untuk menjaga permainan tetap efisien. ini secara keseluruhan menciptakan permainan yang interaktif, di mana sprite klon dihasilkan secara acak dengan Kostum

dan jalur berbeda, bergerak secara dinamis, memberikan sistem poin ketika berhasil dihindari, dan menghentikan permainan jika terjadi tabrakan atau jika *poin* mencapai target kemenangan. Setiap blok kode bekerja saling mendukung untuk menciptakan pengalaman permainan yang menarik dan terorganisir.

6. Pembatas jalan

Gambar 6. https://images.app.goo.gl/LxFQERaTZUTr2iq49. https://images.app.goo.gl/PfRtuzzafbcgeKaZ9.

Permainan ini dirancang dengan tiga logika utama yang saling terintegrasi, yakni kloning objek kendaraan, pergerakan kendaraan, dan modularisasi fungsi. Kloning kendaraan dilakukan dengan memanfaatkan blok *when I start as a clone*, di mana *variabel lane* diatur secara acak dengan rumus untuk menentukan jalur kendaraan (kiri, tengah, atau kanan). Posisi awal kendaraan ditentukan menggunakan koordinat dan , sedangkan kostum kendaraan dipilih secara acak dengan *pick random (1 to 15)*, dan kendaraan ditampilkan menggunakan blok *show* pada *looks*. Kendaraan bergerak vertikal ke bawah dengan rumus dalam loop *repeat until* pada *control*, sementara ukuran kendaraan disesuaikan dengan rumus untuk menciptakan ilusi perspektif. Jika kendaraan menyentuh objek sepeda motor, latar belakang berubah menjadi *Kamu kalah padang gurung*, dan permainan dihentikan pindah ke *control* pilih *stop all*. Klon kendaraan yang melewati batas bawah layar dengan kondisi dihapus menggunakan *delete this clone* dari blok *control*. Pemunculan klon kendaraan diatur dalam blok *when backdrop switches to*, di mana klon baru dibuat secara berkala menggunakan blok create *clone of myself*, dengan interval waktu acak yang ditentukan oleh rumus detik. Untuk mendukung efisiensi manipulasi sprite, digunakan fungsi modular *define goto (x) (y) (costume name),* yang mengatur kostum sprite sementara ke huge, memindahkannya ke koordinat , dan mengembalikan kostumnya ke nilai awal sesuai parameter costume name. Ketiga rumus ini, yang mengintegrasikan konsep pengacakan, perulangan, dan modularisasi, menjadi dasar permainan interaktif dengan pengaturan visual, pergerakan objek, dan kondisi permainan yang dinamis.

7. Mobil Van

Gambar 7. https://images.app.goo.gl/M7LnVUs815cT87mM6. https://images.app.goo.gl/WzZSbce7ErH2C5YS6.

Untuk membangun pembukaan awal program pada tantangan kata kerja agar objek yang tidak seharusnya muncul tetap tersembunyi, pertama pilih blok *when green flag clicked* dari kategori *Event*. Selanjutnya, tambahkan blok *hide* dari kategori *Looks* untuk memastikan semua objek tersembunyi pada awal permainan. Untuk mengatur suara ketika objek tertentu berfungsi sebagai klon, gunakan blok *when I start as a clone* dari *Event*, lalu tambahkan blok *forever* dari *Control* dan sambungkan dengan blok *start sound (car passing)* dari *Sound* untuk memutar suara secara berulang. Untuk pengaturan objek yang ditampilkan berdasarkan perubahan latar, gunakan blok *when backdrop switches to (jalan raya kebun)* dari *Event* yang digabungkan dengan blok *show* dari *Looks*. Kemudian, buat duplikasi blok ini untuk latar *kamu menang* dan *kamu kalah*, tetapi ganti blok *show* menjadi *hide* agar objek tersembunyi pada kondisi tersebut. Pengaturan gerakan horizontal mobil menggunakan variabel *PlayerLane* dengan pemicu blok *when backdrop switches to (jalan raya kebun)*. Dalam loop *forever*, tambahkan operator *and* untuk memeriksa kondisi apakah tombol panah kiri atau kanan ditekan. Jika tombol panah kiri ditekan, tambahkan blok untuk mengurangi nilai *PlayerLane* sebesar -1, sedangkan jika tombol panah kanan ditekan, tingkatkan nilainya sebesar 1. Respons gerakan horizontal ini diperhalus menggunakan blok *wait until* dengan operator *or* untuk menunggu hingga salah satu tombol ditekan sebelum nilai variabel berubah. Untuk gerakan melompat, gunakan blok *when backdrop switches to (jalan raya kebun)* dengan loop *forever*. Dalam loop ini, tambahkan kondisi *if key up arrow pressed* dengan operator *and* untuk memeriksa apakah posisi vertikal *(y position)* bernilai *-100*. Atur variabel *yVelocity* ke 40 untuk memulai lompatan, kemudian kurangi nilainya secara bertahap dengan -6 untuk menciptakan efek gravitasi. Ketika posisi *y* lebih kecil dari *-100*, nilai *yVelocity* diatur menjadi 0, dan posisi *y* dikembalikan ke *-100* untuk menjaga batas bawah gerakan. Akhirnya, kostum sepeda motor disesuaikan dengan jalur yang dipilih pemain menggunakan variabel *PlayerLane*. Gunakan operator *join* untuk menggabungkan nama kostum sesuai nilai *PlayerLane*, sehingga tampilan visual selaras dengan jalur yang dipilih. Pendekatan ini menciptakan pengalaman bermain yang dinamis dan mendukung skenario yang telah dirancang.

8. Palang

Gambar 8. https://images.app.goo.gl/M7LnVUs815cT87mM6. https://images.app.goo.gl/WzZSbce7ErH2C5YS6.

Pada sprite palang pada gambar 8 Menjelaskan bagaimana objek *(palang)* itu muncul dan berapa pada lane dengan rumus pergi ke *event* pilih *backdrop switches to (jalan raya kebun)* lalu pergi ke *kontrol* tarik *forever* masukan lalu ambil *wait (condition) second*, pergi *operation pick random (1.0 to 2.0) second* setelah itu kontrol pilihh *create clone of my self*. Posisi awal kendaraan dalam permainan ditentukan menggunakan koordinat (x, y), sementara kostum kendaraan dipilih secara acak dengan fungsi *pick random (1 to 15).* Kendaraan kemudian ditampilkan di layar melalui blok *show* pada kategori *looks*. Kendaraan bergerak secara vertikal ke bawah dengan memanfaatkan loop *repeat until* yang terdapat pada blok *kontrol*, di mana ukuran kendaraan diatur menggunakan rumus tertentu untuk memberikan efek perspektif visual. Apabila kendaraan menyentuh objek mobil van, permainan akan menampilkan latar belakang *Kamu Kalah* dengan tema kebun, dan permainan dihentikan menggunakan blok *kontrol stop all*. Klon kendaraan yang melampaui batas bawah layar akan dihapus secara otomatis dengan memanfaatkan blok *delete this clone* pada kategori *kontrol*. Proses kloning kendaraan baru diatur menggunakan blok *when backdrop switches to*, yang secara berkala membuat klon baru melalui blok *create clone of myself*. Interval waktu antar klon ditentukan secara acak menggunakan rumus tertentu dalam satuan detik. Untuk mendukung efisiensi pengelolaan sprite, fungsi modular define *goto (x) (y) (costume name)* digunakan. Fungsi ini secara sementara mengubah kostum sprite menjadi *huge*, memindahkannya ke koordinat (x, y), lalu mengembalikan kostum tersebut ke nilai semula sesuai parameter *costume name.* Ketiga pendekatan pengacakan, perulangan, dan modularisasi fungsi diintegrasikan untuk menciptakan mekanisme permainan yang dinamis, meliputi pengaturan visual, pergerakan objek, dan pengelolaan kondisi permainan secara interaktif.

9. Bis Sekolah

Gambar 9. https://images.app.goo.gl/M7LnVUs815cT87mM6. https://images.app.goo.gl/WzZSbce7ErH2C5YS6.

Permainan ini dirancang menggunakan empat coding utama yang saling terintegrasi: kloning objek kendaraan, pengaturan pergerakan kendaraan, dan modularisasi fungsi sprite. Kloning objek dimulai dengan blok *when I start as a clone*, di mana *variabel lane* diatur secara acak ke *nilai -1, 0, atau 1* untuk menentukan jalur kendaraan. Posisi awal kendaraan dihitung dengan rumus *(x = text (lane) times 70)* dan *(y = 180)*, lalu sprite diatur ke kostum *low* dan ditampilkan dengan blok show. Kendaraan bergerak vertikal ke bawah dengan pengurangan posisi (y) sebesar -0,2 dalam loop *repeat until*, sementara ukurannya disesuaikan secara dinamis menggunakan rumus (text (size) = 10 - (text(y position) times 0,04)) untuk menciptakan efek perspektif. Jika kendaraan bersentuhan dengan sepeda motor, latar belakang berubah menjadi *Kamu Kalah* dan seluruh skrip dihentikan menggunakan blok *stop all*. Sebaliknya, jika kendaraan melewati batas layar ((text(y position) < -180)), klon dihapus dengan blok *delete this clone*. Pemunculan klon kendaraan diatur melalui blok *when backdrop switches to* yang digabungkan dengan loop *forever*, dengan waktu tunggu acak antara 1 hingga 2 detik menggunakan rumus *wait (pick random 1.0 to 2.0)*, sebelum membuat klon baru dengan *blok create clone of myself.* Untuk efisiensi manipulasi sprite, digunakan blok khusus *define goto (x) (y) (costume name)* yang mengubah kostum sprite ke *huge*, memindahkannya ke koordinat ((x, y)), lalu mengembalikannya ke kostum asli sesuai parameter *costume name*. Integrasi ketiga logika ini menciptakan mekanisme permainan yang dinamis, responsif, dan interaktif.

Selanjutnya untuk membuat salinan layer yang menyerupai gerbong *bus sekolah*, langkah-langkahnya adalah pertama, buka bagian *My Blocks* dan pilih blok fungsi *spawn train (lane) (length)*. Selanjutnya, tetapkan nilai variabel *lane* menggunakan *set lane to lane* dari *blok goto*, kemudian atur *variabel trainLength* menggunakan *set trainLength* to *length* dari blok yang sama, serta tetapkan *variabel id* ke nilai awal 1 dengan *set id to 1*. Setelah itu, gunakan blok *kontrol repeat, hubungkan dengan perintah goto (length),* tambahkan *create clone of (myself)* untuk membuat salinan objek, dan akhiri dengan memodifikasi nilai *variabel id* menggunakan *change id by 1*. Langkah ini memungkinkan penambahan layer baru berdasarkan *sprite bus sekolah*, khususnya pada kostum ketiga dari objek tersebut.

10. Kata kerja pada lingkaran kuning

Gambar 10. https://images.app.goo.gl/M7LnVUs815cT87mM6. https://images.app.goo.gl/WzZSbce7ErH2C5YS6.

Sistem Scratch yang terdiri dari empat komponen utama yang berfungsi secara berurutan untuk mengatur logika permainan interaktif. Tahap pertama adalah pembuatan klon dengan pengaturan acak dari bagian *Event*, tarik blok *When I start as a clone*, kemudian di bagian *Variables,* pilih *variabel lane* dan atur nilainya menggunakan *set lane to pick random (1) to (6)* untuk menentukan jalur acak tempat klon akan muncul. Selanjutnya, pada bagian *Looks*, tambahkan *switch costume to pick random (1) to (15)* untuk memilih kostum sprite secara acak, masukkan *show* agar klon tampil di layar, dan gunakan *go to front layer* untuk memastikan klon berada di lapisan terdepan. Proses ini bertujuan untuk menghasilkan klon dengan atribut acak, seperti jalur dan kostum, serta memastikan klon terlihat di layar. Pada tahap kedua, pergerakan sprite diatur dengan menggunakan blok *Control*, yaitu dengan memasukkan *repeat until (y position < -180)*, yang membuat sprite bergerak terus-menerus ke bawah hingga keluar layar. Dari bagian *Motion*, tambahkan *change y by (-10)* untuk memindahkan sprite secara vertikal ke bawah, lalu pada bagian *Looks*, tambahkan *set size to (y position * -0.22 + 100)%* untuk menyesuaikan ukuran sprite sesuai posisinya, sehingga menciptakan efek perspektif. Gunakan blok hide dari *Looks* untuk menyembunyikan sprite setelah keluar dari layar. Langkah ini memastikan bahwa sprite bergerak secara dinamis di layar dengan ukuran yang menyesuaikan, menghasilkan efek visual yang realistis. Tahap ketiga melibatkan deteksi tabrakan dan sistem poin dari *Control*, masukkan kondisi *if (y position < -95 AND y position > -105 AND lane = playerLane)* untuk memeriksa apakah sprite bersinggungan dengan pemain (Mobil van). Jika kondisi ini terpenuhi, tambahkan *stop* all dari *Control* untuk menghentikan permainan, lalu dari *Sound*, tambahkan *play sound (Coin) until* done untuk memberikan efek suara ketika pemain berhasil menghindari *sprite,* dan dari *Variables,* masukkan *change Point by (1)* untuk menambahkan poin jika sprite berhasil dihindari. Proses ini mengatur interaksi antara sprite dan pemain, memberikan poin, atau menghentikan permainan jika terjadi tabrakan. Selanjutnya, untuk mendeteksi kondisi kemenangan dan menghapus klon, dari *Control*, tambahkan *if (Point = 35)* untuk memeriksa apakah poin mencapai target kemenangan. Jika terpenuhi, tambahkan *switch backdrop to*

(Kamu menang, Kebun) dari *Looks* untuk mengganti latar belakang dengan tampilan kemenangan, tambahkan *stop all* untuk menghentikan permainan setelah pemain menang, dan akhirnya, tambahkan *delete this clone* dari *Control* untuk menghapus klon setelah proses selesai. Bagian ini memastikan permainan tetap efisien dengan mengelola klon sesuai dengan kondisi kemenangan. Secara keseluruhan, logika ini membangun permainan interaktif di mana sprite klon dihasilkan secara acak dengan kostum dan jalur berbeda, bergerak secara dinamis, memberikan poin ketika berhasil dihindari, serta menghentikan permainan jika terjadi tabrakan atau jika poin mencapai jumlah yang ditargetkan, dengan setiap blok kode saling mendukung untuk menciptakan pengalaman permainan yang menarik dan terstruktur.

11. Pesawat ruang angkasa

Gambar 11. https://images.app.goo.gl/t66vUw4W19icVSB58. https://images.app.goo.gl/rPMtKKmRvS9i8WUAA. https://images.app.goo.gl/9Yy9Ewv4owC53Vos5https://images.app.goo.gl/9Yy9Ewv4owC53Vos5. https://images.app.goo.gl/pWXkUnLZMivobbZb7. https://images.app.goo.gl/gxqdcF2V8o2TRyJBA.

Gambar diatas rumusnya hampir sama dengan objek pada permainan sebelumnya akan tetapi ada tambahan tersebut, jika ingin tidak memunculkan *Pesawat Ruang Angkasa* pada *awal permainan* dengan pergi *event* cari *when greenflag clicked* lalu *looks* tarik *hide.* Selanjutnya jika ingin memunculkan pada permainan ini *(kata sifat) dengan cara pergi ke event* cari *when backdrop switches to* Latar permainan ini *(Galaxy)* setelah itu *pergi ke looks* pilih *show.* Berikutnya pergi ke *event* cari *when i start as a clone lalu ke control* tarik *forever* kasih *sound start sound (car passing)* untuk memunculkan suara. Berikutnya ada dua background *(Kamu Menang)* dan *(Kamu kalah).* Jika muncul tersebut *objek (pesawat ruang angkasa)* di hilangkan dengan cara pergi ke *event* pilih *When backdrop switches to (kamu menang)* lalu ke *looks* cari *hide* dan gabungkan, *duplicate* tersebut menjadi dua dan yang satunya di ganti *(kamu kalah)*

Untuk mengatur gerakan horizontal mobil, digunakan *variabel PlayerLane*, yang diaktifkan melalui blok *when backdrop switches to (Galaxy)* dan terus berjalan di dalam loop *forever*. Dalam mekanisme ini, operator and memastikan apakah tombol panah kiri atau kanan ditekan. Jika tombol panah kiri terdeteksi, nilai PlayerLane akan dikurangi sebanyak -1, sedangkan jika tombol panah kanan ditekan, nilainya akan bertambah 1. Agar transisi ini lebih halus, diterapkan blok *wait until* dengan *operator or,* yang membuat program menunggu aksi tombol sebelum memperbarui nilai variabel tersebut. Untuk gerakan melompat, pengaturan dimulai dengan blok *when backdrop switches to (Galaxy)* yang juga berada dalam *loop forever*. Kondisi melompat dimulai dengan memeriksa apakah tombol panah atas ditekan, sementara posisi vertikal *(y position)* berada pada *nilai -100* menggunakan *operator (condition) and (condition)*. Jika syarat ini terpenuhi, variabel *yVelocity* diberi *nilai 40* untuk memulai lompatan, lalu secara bertahap nilainya berkurang dengan -6 untuk menciptakan efek gravitasi yang realistis. Ketika posisi y kurang dari -100, nilai yVelocity disetel kembali menjadi 0, dan posisi y diatur ke -100 untuk memastikan batas bawah gerakan tidak terlewati. *Kostum Pesawat luar angkasa* kemudian diubah berdasarkan jalur yang dipilih pemain dengan memanfaatkan *variabel PlayerLane. Operator join* digunakan untuk menggabungkan nama kostum sesuai nilai *PlayerLane,* sehingga penampilan visual tetap konsisten dengan jalur permainan. Strategi ini dirancang untuk menciptakan pengalaman bermain yang responsif dan mendukung skenario yang telah dirancang secara dinamis.

12. Meteor

Gambar 12. https://images.app.goo.gl/t66vUw4W19icVSB58. https://images.app.goo.gl/rPMtKKmRvS9i8WUAA. https://images.app.goo.gl/9Yy9Ewv4owC53Vos5https://images.app.goo.gl/9Yy9Ewv4owC53Vos5. https://images.app.goo.gl/pWXkUnLZMivobbZb7. https://images.app.goo.gl/gxqdcF2V8o2TRyJBA.

Pada gambar 12. Ada beberapa rumus, jika ingin memunculkan objek *(meteor)* dengan random dan acak mulai ke event pilih *When backdrop switches to (Galaxy)* pergi ke *control* pilih *wait (condition) second*, pilih *operation* masukan *pick random* didalamnya dan rubah *1.0 to 2.0,* pergi ke *My blocks* pilih *spawn dan tarik spawn train (pick random -1 to 1) (pick random 20 to 40).* Berikutnya pilih *My block* cari *define spawn train* buat menjadi *(lane) (length)* lalu V*ariabel* pilih *set (lane) to (lane) goto, set (trainlength) to (length) goto* dan set (id) to 1, pergi ke *control* pilih repeat tambahka length (goto) lalu balik control ambil create clone my of *(myself)* dan ganti blok variabel pilih change *(id) by 1*. Kloning objek dimulai dengan blok *when I start as a clone,* di mana variabel lane diatur secara acak ke nilai *-1, 0,* atau *1* untuk menentukan jalur kendaraan. Posisi awal kendaraan dihitung dengan rumus $x = lane \times 70$ $x = lane \times 70$ dan $y = 180$ $y = 180$, sehingga kendaraan dimulai dari posisi yang sesuai dengan jalurnya. Sprite kendaraan diatur ke kostum *low* dan ditampilkan menggunakan blok *show*. Selanjutnya, kendaraan bergerak vertikal ke bawah dengan mengurangi posisi y y *sebesar -0,2* dalam loop *repeat until.* Ukurannya juga disesuaikan secara dinamis menggunakan rumus size $= 10 - (y\ position \times 0{,}04)$ size $=10 - (y\ position \times 0{,}04)$, menciptakan efek perspektif. Jika kendaraan bersentuhan dengan *Pesawat luar angkasa*, latar belakang berubah menjadi *Kamu Kalah,* dan seluruh skrip dihentikan dengan *stop all.* Sebaliknya, jika kendaraan melewati batas layar $(y\ position < -180\ y\ position < -180)$. Lalu jika objek *(pesawat luar angkasa)* mengenai *meteor* akan berganti background menjadi *(kamu kalah galaxy)* dan kasih *stop all* pada *control*
dan *kasih detete this clone.*

13. Planet

Gambar 13. https://images.app.goo.gl/t66vUw4W19icVSB58. https://images.app.goo.gl/rPMtKKmRvS9i8WUAA. https://images.app.goo.gl/9Yy9Ewv4owC53Vos5https://images.app.goo.gl/9Yy9Ewv4owC53Vos5. https://images.app.goo.gl/pWXkUnLZMivobbZb7. https://images.app.goo.gl/gxqdcF2V8o2TRyJBA.

Program ini dirancang untuk mengelola kloning kendaraan dalam sebuah game dengan logika yang terstruktur. Ketika klon dimulai, *variabel Lane* diatur secara acak menggunakan *lane = pick random(−1,1) text (lane) = text pick random (-1, 1),* yang menentukan jalur Planet, serta kostum dipilih dengan *costume = pick random(1,15) text (costume) = text (pick random) (1, 15).* Posisi awal kendaraan ditentukan menggunakan $x = lane \times 70$ $x =$ *text (lane) times 70 dan* $y=180$ $y = 180$, memastikan kendaraan berada pada koordinat tertentu sesuai jalur. Pergerakan kendaraan ke bawah diatur dalam loop *repeat until* dengan posisi vertikal yy diperbarui setiap iterasi menggunakan $y = y - 0.2$ $y = y - 0.2$, sementara ukuran sprite disesuaikan secara dinamis berdasarkan posisi vertikal menggunakan rumus *size = 10 − (y×0.04) text (size) = 10 - (y times 0.04),* untuk menciptakan efek perspektif. Jika

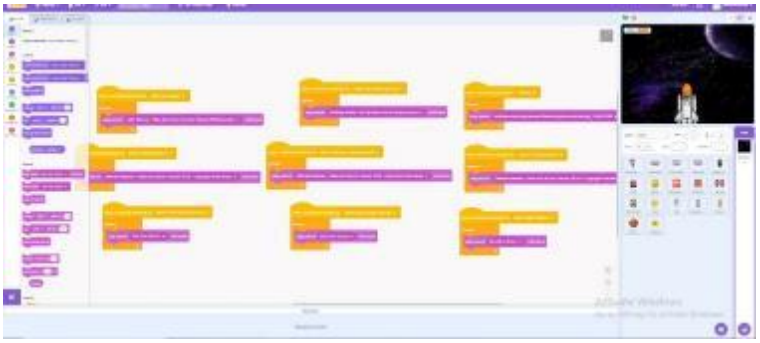
kendaraan menyentuh sprite pesawat ruang angkasa, skrip menghentikan seluruh proses melalui blok `stop all` setelah mengubah latar belakang menjadi *Kamu Kalah*. Sebaliknya, jika kendaraan keluar dari layar dengan kondisi *y<−180y < -180*, klon akan dihapus menggunakan blok `delete this clone`. Klon baru dibuat melalui loop `forever` dengan waktu tunggu acak antara 1,5 hingga 2 detik menggunakan rumus *wait=pick random (1.5,2.0) text (wait) = \text (pick random) (1.5, 2.0),* kemudian memanggil blok `create clone of myself`. Selain itu, blok khusus `define goto(x, y, costume name)` digunakan untuk memindahkan sprite ke koordinat *(x,y)(x, y)* menggunakan kostum sementara, misalnya *huge,* sebelum mengembalikan sprite ke kostum asli sesuai parameter `costume name`. Program ini secara keseluruhan dirancang untuk mensimulasikan kendaraan yang bergerak pada jalur acak dengan efek perspektif, menciptakan tantangan bagi pemain untuk menghindari tabrakan, sekaligus memastikan gameplay berjalan lancar melalui pengaturan klon yang efisien.

14. Bintang

Gambar 14. https://images.app.goo.gl/t66vUw4W19icVSB58. https://images.app.goo.gl/rPMtKKmRvS9i8WUAA. https://images.app.goo.gl/9Yy9Ewv4owC53Vos5https://images.app.goo.gl/9Yy9Ewv4owC53Vos5. https://images.app.goo.gl/pWXkUnLZMivobbZb7. https://images.app.goo.gl/gxqdcF2V8o2TRyJBA.

Program pada gambar Scratch ini mengatur kloning kendaraan menggunakan berbagai rumus dan logika. Jalur kendaraan ditentukan secara acak melalui *(text (ane) = text (pick random) (-1, 1)),* yang memetakan posisi horizontal ke nilai (x = \text{lane} \times 70) dengan koordinat vertikal awal *(y = 180).* Kostum kendaraan dipilih secara acak menggunakan *(text (costume) = text(pick random) (1, 15)).* Kendaraan bergerak ke bawah secara vertikal dengan rumus *(y = y - 0.2)* dalam loop, sementara ukuran sprite diatur secara dinamis berdasarkan posisi vertikal menggunakan *(text (size) = 10 - (y times 0.04)),* menciptakan efek perspektif. Kondisi permainan mencakup pengecekan tabrakan melalui (text (touching) (text (pesawat luar angkasa)) yang memicu perubahan latar belakang menjadi *Kamu menang Galaxy,* memainkan suara *Coin*, dan mengakhiri skrip ini jika mengenai akan muncul point dalam mengambil kata benar. Kendaraan yang keluar dari layar bawah dengan kondisi *(y < -180)* dihapus menggunakan blok *delete this clone* untuk menjaga efisiensi. Selain itu, klon kendaraan baru dibuat dengan jeda acak menggunakan *(text (wait) = text (pick random) (0.5, 1.5)).* Program ini juga menggunakan blok khusus *goto(x, y, costume name)* untuk mengatur posisi dan kostum sprite secara efisien. Semua elemen ini dirancang untuk menghasilkan gameplay yang dinamis, menantang, dan realistis.

15. Background pada scratch



Gambar 15. https://images.app.goo.gl/tsXdKbZh2opRjojY7. https://images.app.goo.gl/PfRtuzzafbcgeKaZ9. https://images.app.goo.gl/9Yy9Ewv4owC53Vos5https://images.app.goo.gl/9Yy9Ewv4owC53Vos5.

Pada scratch gambar ini cukup simpel hanya di ganti background dan sound yang menyelimuti fase permainan tersebut misaln pilih blok event lalu tarik when backdrop switches to jalan raya padang gurun/ kebun/ galaxy) lalu pergi control tarik forever lalu pilih sound yang ingin dimasukan Play sound (sound yang ingin digunakan). Lalu duplicate menjadi tiga jalan raya padan gurun, tiga jalan raya kebun, dan galaxy, ganti di setiap sound yang ingin digunakan pada background ini, menjadi sepuluh background dari awal permainan sampai dengan background galaxy.

SIMPULAN

Desain game Petualangan Edukasi Menangkap Kata berbasis Scratch untuk meningkatkan pemahaman terhadap kata benda, kata kerja, dan kata sifat siswa sekolah dasar memiliki blok kode pada 1) Awal permainan, 2) Kata (benda, kerja, sifat), 3) Sepeda motor, 4) Mobil, 5) Kata benda pada lingkaran kuning, 6) Pembatas jalan, 7) Mobil van, 8) Palang, 9) Bis sekolah, 10) Kata kerja pada lingkaran kuning, 11) Pesawat ruang angkasa, 12) Meteor, 13) Pelanet, 14) Bintang, 15) Background pada scratch.

REFERENSI

Al Ayubi, M. Z. F., & Sulistyowati, H. (2023). Kemenarikan Pemilihan Kata Pada Iklan Online Shop Aplikasi Shopee Analisis Wacana Persuasi. Journal on Education, 5(4), 17104-17114. https://doi.org/10.31004/joe.v5i4.4059.

Albab, U. (2021). Perencanaan Pendidikan dalam Manajemen Mutu Terpadu Pendidikan Islam. Jurnal Pancar (Pendidik Anak cerdas dan Pintar), 5(1), 119-126. https://doi.org/10.52802/pancar.v5i1.104.

Aminingtyas, M., & Wardhani, J. D. (2023). Hubungan Minat dan Motivasi Belajar Berbasis Portal Rumah Belajar terhadap Hasil Belajar Kognitif Anak. Murhum: Jurnal Pendidikan Anak Usia Dini, 4(1), 590-601. https://doi.org/10.37985/murhum.v4i1.268.

Ardiansyah, T. (2020). Kreativitas dan inovasi dalam berwirausaha. Jurnal Usaha, 1(2), 19-25. https://doi.org/10.30998/juuk.v1i2.503.

Arsyad, M. N. (2018). Penerapan Media Pembelajaran Berbasis Multimedia Interaktif Terhadap Mahasiswa IKIP Budi Utomo Malang. Agastya: Jurnal Sejarah Dan Pembelajarannya, 8(2), 188-198. https://doi.org/10.25273/ajsp.v8i2.2702.

Badrul, M. (2021). Penerapan Metode Waterfall Untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang. PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer, 8(2), 57-52. https://doi.org/10.30656/prosisko.v8i2.3852.

Bashari, F. R., Alfarizi, M. R., Sitanggang, H. R., & Kurniawan, H. (2024). Rancang Bangun Toko Online Berbasis Web Pada Zelay Store Menggunakan Metode Waterfall: Design and Build a Web-Based Online Store at Zelay Store Using the Waterfall Method. Jurnal Komputer Teknologi Informasi dan Sistem Informasi (JUKTISI), 3(1), 673-680. https://doi.org/10.62712/juktisi.v3i1.205.

Chaerunnisa, N. A., & Bernard, M. (2021). Analisis minat belajar siswa sekolah dasar pada pembelajaran Matematika dengan menggunakan media Scratch. JPMI (Jurnal Pembelajaran Matematika Inovatif), 4(6), 1577-1584. https://doi.org/10.22460/jpmi.v4i6.p%25p.

Dhaifullah, I. R., Salsabila, A. A., & Yaqin, M. A. (2022). Survei Teknik Pengujian Software. Journal Automation Computer Information System, 2(1), 31-38. https://doi.org/10.47134/jacis.v2i1.42.

Djafar, C. (2020). KAJIAN DIKSI DAN GAYA BAHASA METAFORA PADA PUISI ININAWA KARYA LAKON SANG KELANA MODIES PALOPO. Jurnal Andi Djemma| Jurnal Pendidikan, 3(2), 1-7. https://doi.org/10.35914/jad.v3i2.445.

Elsola, D. A. N., Cahyani, B. H., Havifah, B., & Nisa, A. F. (2023). Penerapan Model Sole dan Pemanfaatan Scratch untuk Meningkatkan Motivasi Belajar Siswa SD Negeri Selo. Pendas: Jurnal Ilmiah Pendidikan Dasar, 8(2), 2030-2045. https://doi.org/10.23969/jp.v8i2.9201.

Fachri, B., & Surbakti, R. W. (2021). Perancangan Sistem Dan Desain Undangan Digital Menggunakan Metode Waterfall Berbasis Website (Studi Kasus: Asco Jaya). Journal Of Science And Social Research, 4(3), 263-267. https://doi.org/10.54314/jssr.v4i3.692.

Hakim, M. W. A., & Kasiyun, S. (2024). PENERAPAN SCRATCH SEBAGAI MEDIA PEMBELAJARAN INOVATIF PADA MATERI BANGUN RUANG SISWA KELAS V SEKOLAH DASAR. Jurnal Review Pendidikan dan Pengajaran (JRPP), 7(4), 12487-12493. https://doi.org/10.31004/jrpp.v7i4.34055.

Hardiansyah, B., Armin, A. P., & Rahmadi, A. A. (2023). Implementasi aplikasi game menggunakan Scratch dalam meningkatkan hasil belajar dan motivasi belajar siswa. J-ABDI: Jurnal Pengabdian kepada Masyarakat, 3(4), 707-716. https://doi.org/10.53625/jabdi.v3i4.6464.

Hardianto, M., Widayati, W., & Sucipto, S. (2017). Diksi dan gaya bahasa pada naskah pidato presiden soekarno. Jurnal Ilmiah Fonema: Jurnal Edukasi Bahasa dan Sastra Indonesia, 4(2). https://doi.org/10.25139/fn.v4i2.761.

Hariadi, H., Mahfuz, M., Nopiana, R., Saputra, E., Thuhir, M., Daniyantara, D., & Suryadi, L. E. (2023). Penerapan metode interaktif dalam meningkatkan

keaktifan siswa pada pembelajaran penjaskes. Jurnal Porkes, 6(2), 837-853. https://doi.org/10.29408/porkes.v6i2.23746.

Harsiwi, U. B., & Arini, L. D. D. (2020). Pengaruh pembelajaran menggunakan media pembelajaran interaktif terhadap hasil belajar siswa di Sekolah Dasar. Jurnal Basicedu, 4(4), 1104-1113. https://doi.org/10.31004/basicedu.v4i4.505.

Hidayah, S., Wahyuni, S., & Ani, H. M. (2017). Penggunaan Media Pembelajaran Interaktif dengan Aplikasi Adobe Flash CS6 Untuk Meningkatkan Motivasi Belajar Pada Kompetensi Dasar Menganalisis Peran, Fungsi dan Manfaat Pajak (Studi Kasus Siswa Kelas XI IPS 1 MAN 1 Jember Semester Genap Tahun Ajaran 2016. JURNAL PENDIDIKAN EKONOMI: Jurnal Ilmiah Ilmu Pendidikan, Ilmu Ekonomi dan Ilmu Sosial, 11(1), 117-123. https://doi.org/10.19184/jpe.v11i1.5012.

Jannah, D. M., Hidayat, M. T., Ibrahim, M., & Kasiyun, S. (2021). Pengaruh Kebiasaan Belajar dan Motivasi Belajar terhadap Prestasi Belajar Siswa di Sekolah Dasar. Jurnal Basicedu, 5(5), 3378-3384. https://doi.org/10.31004/basicedu.v5i5.1350.

Jannah, U. R., Putra, F. P. E., Hafsi, A. R., & Basri, H. (2021). Pengembangan sekolah inklusi dengan pemanfaatan media visual scratch dan alat manipulatif. Wikrama Parahita: Jurnal Pengabdian Masyarakat, 5(1), 89-96. https://doi.org/10.30656/jpmwp.v5i1.2653.

Martanti, A. P., Hardyanto, W., & Sopyan, A. (2013). Pengembangan media animasi dua dimensi berbasis java scratch materi teori kinetik gas untuk meningkatkan pemahaman konsep siswa SMA. UPEJ Unnes Physics Education Journal, 2(2). https://doi.org/10.15294/upej.v2i2.2661.

Mukhlis, I. R. (2022). Literature Review Pada Teknik Pendeteksi Ambiguitas Leksikal dalam Software Requirements Specification. http://eprints.perbanas.ac.id/id/eprint/10424.

Munthe, I. R. (2017). Perancangan Sistem Informasi Pengarsipan Data Penduduk Pada Kantor Camat Bilah Hulu Kabupaten Labuhan Batu Dengan Metode System Develovment Life Cycle (Sdlc). Informatika, 5(1), 22-31. https://doi.org/10.36987/informatika.v5i1.666.

Nasrullah, M., Angresti, N. D., Suryawan, S. H., & Mahananto, F. (2021). Requirement Engineering terhadap Virtual Team pada Proyek Software Engineering. Journal of Advances in Information and Industrial Technology, 3(1), 1-10. https://doi.org/10.52435/jaiit.v3i1.79.

Prasetya, A. D. A. (2019). Analisis kesalahan ejaan dan pilihan kata pada surat dinas di STKIP Al Hikmah Surabaya. Lingua Franca: Jurnal Bahasa, Sastra, dan Pengajarannya, 3(1), 120-127. https://doi.org/10.30651/lf.v3i1.2377.

Pratiwi, A. P., & Bernard, M. (2021). Analisis minat belajar siswa kelas v sekolah dasar pada materi satuan panjang dalam pembelajaran menggunakan media scratch. JPMI (Jurnal Pembelajaran Matematika Inovatif), 4(4), 891-898. https://doi.org/10.22460/jpmi.v4i4.p%25p.

Pratiwi, I., Anardani, S., & Putera, A. R. (2023). Rancang Bangun Sistem Informasi Penjadwalan Mata Pelajaran dengan Metode Waterfall. JDMIS: Journal of

Data Mining and Information Systems, 1(1), 20-28. https://doi.org/10.54259/jdmis.v1i1.1513.

Qurin, M. T., Wijayanti, K. D., Fathori, A. R., Sukma, H. F., Setiawan, H., Pratama, K. H., ... & Khoiriyah, N. H. M. (2024). Pelatihan Coding Berbasis Project Based Learning (PjBL) Menggunakan Platform Scratch untuk Sekolah Dasar. Society: Jurnal Pengabdian Masyarakat, 3(5), 283-291. https://doi.org/10.55824/jpm.v3i5.437.

Salleh, S. F., Yahya, Y., Subet, M. F., & Daud, M. Z. (2020). Analisis semantik leksikal dalam novel Sangkar karya Samsiah Mohd. Nor. Asian People Journal (APJ), 3(1), 45-63. https://doi.org/10.37231/apj.2020.3.1.144.

Shalikhah, N. D. (2017). Media pembelajaran interaktif lectora inspire sebagai inovasi pembelajaran. Warta Lpm, 20(1), 9-16. https://doi.org/10.23917/warta.v19i3.2842.

Sitanggang, P. P., Permatasari, I., Rhamadan, S., & Amsury, F. (2023). Penerapan Metode Waterfall Pada Sistem Informasi Pendaftaran dan Pembayaran Membership Komunitas United Indonesia. Jurnal Komputer Antartika, 1(4), 189-198. https://doi.org/10.70052/jka.v1i4.197.

Sudihartinih, E., Novita, G., & Rachmatin, D. (2021). Desain media pembelajaran matematika topik luas daerah segitiga menggunakan aplikasi scratch. Jurnal Cendekia: Jurnal Pendidikan Matematika, 5(2), 1390-1398. https://doi.org/10.31004/cendekia.v5i2.643.

Supardi, A. (2014). Penggunaan Multimedia Interaktif Sebagai Bahan Ajar Suplemen Dalam Peningkatan Minat Belajar. Jurnal Ilmiah Pendidikan Dasar, 1(2), 161-167. http://dx.doi.org/10.30659/pendas.1.2.161=167.

Suparman, N. F. N. (2019). Inovasi Leksikal Bahasa Wotu. Ranah: Jurnal Kajian Bahasa, 8(2), 219-236. https://doi.org/10.26499/rnh.v8i2.1282..

Suryaningsih, L. (2017). Kesalahan diksi dan kalimat pengisi suara Acara infotainment insert. Cendekia: Jurnal Pendidikan dan Pembelajaran, 11(1), 79-92. https://doi.org/10.30957/cendekia.v11i1.253.

Tresnawati, F. (2023). DIKSI DAN MAJAS DALAM KUMPULAN PUISI TENTANG PERJUANGAN KARYA TAUFIK ISMAIL: KAJIAN STILISTIKA. PUSTAKA: Jurnal Bahasa dan Pendidikan, 3(1), 01-05. https://doi.org/10.56910/pustaka.v3i1.275.

Wijaya, G. A. A., Ikhwan, A., & Putri, R. A. (2023). Sistem Informasi Manajemen Aset Tetap Menggunakan Metode Waterfall. Resolusi: Rekayasa Teknik Informatika dan Informasi, 3(6), 269-278. https://doi.org/10.30865/resolusi.v3i6.992.

Winarko, H. B. (2011). PERAN HUBUNGAN INTERAKTIF TERHADAP PERTUMBUHAN INDUSTRI E-COMMERCE DAN EVOLUSI INDUSTRI JEJARING SOCIAL MEDIA. Journal of Management and Business Review, 8(2). https://doi.org/10.34149/jmbr.v8i2.84.

Yulistiana, E., Sumarlam, S., & Satoto, S. (2019). Mengungkap penggunaan diksi lirik lagu tarlingdut karya Abdul Adjib: Kajian stilistika. KEMBARA: Jurnal Keilmuan Bahasa, Sastra, dan Pengajarannya, 5(1), 53-62. https://doi.org/10.22219/kembara.v5i1.6400.

Yusron, R. D. R., & Huda, M. M. (2021). Analisis Perancangan Sistem Informasi Perpustakaan Menggunakan Model Waterfall Dalam Peningkatan Inovasi Teknologi. Journal Automation Computer Information System, 1(1), 26-36. https://doi.org/10.47134/jacis.v1i1.4.