



## Prosiding

### Seminar Nasional

Unit Kegiatan Mahasiswa Penalaran dan Riset

IKIP PGRI Bojonegoro

Tema “Eksplorasi Penalaran dalam Riset untuk Meningkatkan Kualitas Publikasi Ilmiah”



---

# Desain Game Menangkap Kata Sifat Berbasis Scratch untuk Melatih Kemampuan Siswa Sekolah Dasar

Henriska Nitra Amelia<sup>1</sup>(✉), Tri Inayah Maula<sup>2</sup>, Cahyo Hasanudin<sup>3</sup>

<sup>1,2,3</sup>Pendidikan Bahasa dan Sastra Indonesia, IKIP PGRI Bojonegoro, Indonesia

[nitraamelia2019@gmail.com](mailto:nitraamelia2019@gmail.com)

**abstrak**— Kata sifat adalah kata yang digunakan untuk mendeskripsikan atau memberikan sifat dan karakteristik pada kata benda, seperti ukuran, warna, atau bentuk. Tujuan untuk melatih kemampuan mengidentifikasi kata sifat siswa sekolah dasar. Metode penelitian ini adalah SDLC dengan model waterfall. Model ini merupakan pendekatan tradisional dalam pengembangan perangkat lunak yang dilakukan secara sistematis dan berurutan. Hasil penelitian ini adalah desain game kata sifat berbasis scratch untuk melatih kemampuan mengidentifikasi kata sifat siswa sekolah dasar memiliki blok kode 1) Sprite Mangkok, 2) Tombol 2, 3) Tombol 3, 4) Tombol 4, 5) Dinosaur karakter utama pada level 1, 6) Benar level 2, 7) Salah level 1, 8) Monyet karakter utama pada level 2, 9) Benar level 2, 10) Salah level 2, 11) Kupu-kupu karakter utama level 3, 12) Benar level 3, 13) Salah level 3. Simpulan dalam penelitian ini terdapat 13 desain permainan kata sifat berbasis scratch untuk melatih kemampuan mengidentifikasi kata sifat siswa sekolah dasar.

**Kata kunci**— Kata sifat, Scratch, Siswa Sekolah Dasar

**abstract**— Adjectives are words used to describe or give properties and characteristics to nouns, such as size, color, or shape. The aim is to train the ability to identify adjectives of elementary school students. This research method is SDLC with waterfall model. This model is a traditional approach in software development that is done systematically and sequentially. The result of this research is a scratch-based adjective game design to train the ability to identify adjectives of elementary school students has code blocks 1) Sprite Bowl, 2) Button 2, 3) Button 3, 4) Button 4, 5) Dinosaur main character at level 1, 6) True level 2, 7) Incorrect level 1, 8) Monkey main character at level 2, 9) Correct level 2, 10) Incorrect level 2, 11) Butterfly main character level 3, 12) Correct level 3, 13) Incorrect level 3. The conclusion in this research is that there are 13 scratch-based adjective game designs to train elementary school students' adjective identification skills.

**Keywords**— Adjectives, Scratch, Primary School Students

## PENDAHULUAN

Kata sifat adalah kata yang digunakan untuk mendeskripsikan atau memberikan sifat dan karakteristik pada kata benda, seperti ukuran, warna, atau bentuk (Zandroto dkk., 2024) dan berfungsi untuk menjelaskan kualitas atau kondisi dari seseorang atau sesuatu sehingga memberi informasi tambahan pada kata benda yang diterangkan (Kartika & Budaya, 2017) serta merupakan jenis kata yang menjelaskan karakteristik atau keadaan objek dalam kalimat sehingga membuat deskripsi lebih spesifik, seperti "panjang," "berani," atau "halus." (Abdussamad, 2022) disamping itu kata sifat atau adjektiva merupakan jenis kata yang digunakan untuk menjelaskan atau memberikan keterangan tentang sifat, keadaan, atau karakteristik suatu benda, orang, tempat, atau hal (Rosyidah dkk., 2021). Kata sifat biasanya berfungsi untuk memberikan penjelasan tambahan dalam sebuah kalimat, baik secara kualitas, kuantitas, atau keadaan lainnya (Arianti dkk., 2020). Kata sifat dibagi menjadi dua, yaitu kata sifat deskriptif dan kata sifat pembatas.

Kata sifat deskriptif dan kata sifat pembatas (Rumokoy, 2018). Kata deskriptif memberikan gambaran rinci dengan memberikan rincian spesifik tentang karakteristik suatu benda atau orang, seperti warna, bentuk, ukuran, atau kualitas lainnya yang bisa dilihat atau dirasakan, seperti pada contoh "mobil merah" atau "tas besar" (Rahman, 2017) selain itu, kata sifat deskriptif juga menggambarkan sifat fisik atau emosional, yang membantu pembaca atau pendengar memahami kondisi atau perasaan dari subjek yang dibicarakan, contohnya "senyum ramah" atau "suasana damai" (Liliweri, 2017) di samping itu kata sifat deskriptif juga berperan untuk memperkaya deskripsi dalam kalimat, sehingga menjadikannya lebih menarik dan informatif, serta memberi warna atau nuansa lebih pada objek atau subjek yang dibahas, seperti "pemandangan indah" atau "bunga harum" (Widarmanto, 2017). Kata sifat pembatas digunakan untuk menunjukkan jumlah atau kuantitas dari suatu benda atau orang, seperti pada contoh "beberapa," "banyak," atau "sedikit," yang memberi batasan pada jumlah yang dimaksud (Jalaluddin, 2024) dan juga berfungsi untuk membatasi ruang lingkup atau spesifikasi objek yang dimaksud, seperti kata "setiap," "semua," atau "beberapa," yang menunjukkan batasan objek yang digambarkan (Suyanto, 2015) selain itu, kata sifat pembatas juga digunakan untuk menunjukkan kepemilikan atau asal dari sesuatu, misalnya pada kata "miliknya," "punya," atau "asal," yang memberikan batasan terkait siapa yang memiliki atau dari mana benda tersebut berasal (Azzam, 2022). Adapun jenis-jenis kata sifat kualitatif, kuantitatif, dan demonstratif.

Kata sifat kualitatif digunakan untuk menggambarkan sifat atau kualitas suatu benda atau orang, baik itu sifat fisik maupun emosional, seperti pada contoh "baik," "cerdas," atau "cantik," (Wahyuni & Harun, 2018) dan juga berfungsi untuk memberikan penjelasan tentang keadaan atau kondisi sesuatu, baik itu kondisi fisik, emosional, atau situasional, seperti "sehat," "bahagia," atau "lelah" (Mashar, 2015) selain itu, kata sifat kualitatif menggambarkan ciri atau karakteristik tertentu dari

suatu objek atau subjek yang sedang dibicarakan, seperti "tinggi," "gelap," atau "lembut" (Sadli, 2010).

Kata sifat kuantitatif digunakan untuk menunjukkan jumlah atau kuantitas dari suatu benda atau orang, baik secara spesifik atau tidak, seperti pada contoh "beberapa," "banyak," atau "sedikit" (Olivia & Nirmalasari, 2022) dan juga berfungsi untuk menunjukkan perbandingan jumlah antara dua objek atau subjek, seperti "lebih banyak" atau "lebih sedikit" (Winarno, 2013) selain itu, kata sifat kuantitatif dapat menggambarkan tingkat kepadatan atau frekuensi, seperti pada kata "selalu," "kadang-kadang," atau "jarang" (Nuzuli, 2022).

Kata sifat demonstratif digunakan untuk menunjukkan atau menegaskan objek tertentu yang sedang dibicarakan dalam kalimat, seperti pada contoh "ini," "itu," atau "tersebut" (Sulaksono, 2019) dan juga mengacu pada jarak atau lokasi objek terhadap pembicara, seperti "ini" (dekat) dan "itu" (jauh) (Damayanti, 2015) selain itu, kata sifat demonstratif memberi penekanan pada objek yang dimaksud, misalnya "buku ini" atau "rumah itu," yang membantu membedakan objek yang sedang dibicarakan (Hansopaheluwakan dkk., 2024).

Scratch adalah bahasa pemrograman visual yang dirancang untuk memudahkan pengguna, terutama anak-anak dan pemula, dalam membuat proyek seperti animasi, permainan, serta cerita interaktif (Afriani, 2022) dengan menggunakan blok-blok kode yang dapat disusun tanpa harus mempelajari sintaks pemrograman yang rumit, Scratch berfungsi sebagai platform pembelajaran yang bertujuan untuk mengenalkan konsep dasar pemrograman dan logika berpikir (Isnaini dkk., 2021) selain itu, platform ini dikembangkan oleh MIT Media Lab, yang mengusung antarmuka drag-and-drop yang intuitif, sehingga memungkinkan pengguna membuat proyek interaktif dengan mudah dan efektif (Nabila dkk., 2024). Scratch memberikan banyak manfaat.

Scratch bisa dimanfaatkan untuk membuat game edukasi, yang membantu siswa belajar dengan cara yang interaktif dan menyenangkan, sekaligus mempermudah pemahaman materi melalui permainan seperti kuis sejarah atau tebak kata (Perkasa & Wantoro, 2024) selain itu, Scratch juga dapat meningkatkan kreativitas siswa dalam seni digital, karena mereka dapat menciptakan animasi, cerita, atau simulasi visual yang mendukung pembelajaran seni, yang pada gilirannya mendorong mereka untuk lebih kreatif dalam menyampaikan ide Lebih jauh lagi (Utami, 2024) bersamaan dengan itu scratch juga bermanfaat dalam mengembangkan logika pemrograman, di mana siswa dilatih untuk menggunakan blok kode dalam menyelesaikan masalah sederhana dan melatih pola pikir komputasional (Mufidah & Majid, 2024). Scratch memiliki kelebihan dan kekurangan.

Kelebihan Scratch menawarkan antarmuka visual yang mudah dipahami, yang membuatnya sangat cocok bagi pemula, terutama anak-anak, karena mereka dapat belajar pemrograman tanpa harus menulis kode secara langsung (Maharani dkk., 2020) selain itu, Scratch tersedia secara gratis dan mudah diakses, baik melalui versi

online maupun offline, sehingga pengguna tidak perlu khawatir tentang biaya tambahan (Peniarsih, 2021) lebih lanjut, Scratch juga mendorong kreativitas dan kolaborasi, karena pengguna dapat membuat berbagai proyek seperti animasi, game, atau cerita interaktif, serta membagikan proyek mereka dengan komunitas global (Setiawan dkk., 2023).

Namun, ada beberapa kekurangan Scratch, seperti kurangnya dukungan untuk pemrograman tingkat lanjut karena keterbatasan fitur yang tersedia (Priharsanto, 2024) di samping itu, pengguna versi online harus bergantung pada koneksi internet, yang bisa menjadi masalah di tempat-tempat dengan akses internet yang terbatas (Hamzah, 2015) terakhir, Scratch juga memiliki keterbatasan dalam hal kompatibilitas dengan perangkat lain, sehingga menjadi kurang ideal untuk proyek-proyek yang memerlukan integrasi dengan perangkat keras atau platform lain (Julianti, 2021)

Siswa sekolah dasar merupakan anak-anak yang tengah mengikuti jenjang pendidikan formal pada tingkat sekolah dasar, yang menjadi bagian dari pendidikan dasar wajib di Indonesia (Asiah, 2018) umumnya, mereka berusia antara 6 hingga 12 tahun, sesuai dengan rentang usia belajar yang telah ditetapkan dalam sistem pendidikan nasional. Pada tahap ini, siswa difokuskan untuk menguasai kemampuan dasar, seperti membaca, menulis, dan berhitung (Kurniawan, 2013), yang dikenal sebagai fondasi penting dalam proses pembelajaran. Selain itu, pendidikan di tingkat sekolah dasar juga bertujuan untuk mengembangkan karakter positif, membentuk moral yang baik, serta meningkatkan kemampuan sosial siswa agar dapat berinteraksi dengan lingkungan sekitar secara efektif (Hakim, 2023). Dengan demikian, siswa sekolah dasar memainkan peran penting dalam mempersiapkan generasi muda untuk jenjang pendidikan berikutnya dan kehidupan bermasyarakat. Setiap siswa sekolah dasar memiliki karakteristik yang berbeda-beda.

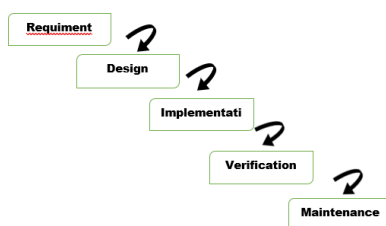
Siswa sekolah dasar memiliki karakteristik khas yang mencakup perkembangan fisik, kognitif, dan sosial-emosional. Secara fisik, mereka sangat aktif, memiliki energi tinggi, serta mulai menguasai kemampuan motorik halus dan kasar, seperti menulis atau bermain olahraga (Rismayanthi, 2013) sementara itu, dari segi kognitif, mereka berada pada tahap operasional konkret sehingga mulai mampu berpikir logis, meskipun masih terbatas pada hal-hal konkret, dan memiliki rasa ingin tahu yang tinggi (Holis, 2017) selain itu, secara sosial-emosional, mereka cenderung membentuk hubungan erat dengan teman sebaya, menyukai kegiatan berkelompok, serta mulai belajar memahami aturan sosial dan kerja sama, walaupun mereka masih sering mencari perhatian dari orang dewasa (Muzzamil, 2021). Gaya belajar siswa sekolah dasar sangat beragam.

Gaya belajar utama siswa sekolah dasar adalah visual, auditori, dan kinestetik. Siswa dengan gaya belajar visual lebih mudah memahami informasi melalui gambar, diagram, atau tulisan, serta cenderung mengingat warna dan bentuk dengan baik (Fendrik dkk., 2022) sementara itu, siswa auditori lebih efektif belajar melalui mendengarkan penjelasan atau diskusi, sehingga mereka lebih cepat memahami

materi melalui instruksi lisan (Supit dkk., 2023) selain itu, siswa dengan gaya belajar kinestetik cenderung belajar lebih baik dengan melakukan aktivitas fisik atau praktik langsung, seperti eksperimen atau permainan, yang memungkinkan mereka belajar melalui pengalaman (Mustafida, 2013).

## METODE PENELITIAN

Penelitian ini menggunakan metode SDLC dengan model waterfall. Model ini merupakan pendekatan tradisional dalam pengembangan perangkat lunak yang dilakukan secara sistematis dan berurutan (Sandy & Lazuardi, 2024) setiap tahap dalam metode ini harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya, tanpa adanya proses pengulangan tanpa revisi (Widiyanto, 2018) model waterfall ini cocok digunakan untuk proyek yang memiliki kebutuhan yang telah jelas dan terdefinisi sejak awal. Pendekatan ini memberikan struktur yang kuat karena setiap tahap memiliki dokumentasi yang lengkap, sehingga memudahkan pengembang perangkat secara keseluruhan (Gultom & Murpratomo, 2018). Pendekatan ini terdiri dari lima tahapan utama yang dilakukan secara terorganisasi sebagaimana berikut.



Gambar 1. Tahapan Waterfall

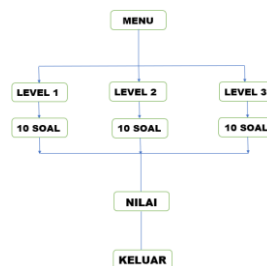
### 1. Requirement (Kebutuhan)

Requirement adalah tahap proses pengumpulan, analisis, dan dokumentasi kebutuhan pengguna atau sistem, yang mencakup identifikasi kebutuhan fungsional maupun non-fungsional sebagai dasar pengembangan perangkat lunak (Pertwi & Kurniawan, 2016) selain itu, tahap ini juga bertujuan untuk memastikan sistem dapat memenuhi kebutuhan pengguna secara efektif sesuai spesifikasi yang telah ditentukan (Harimurti & Udariansyah, 2023) lebih lanjut, requirement membantu mendeskripsikan kebutuhan pengguna secara rinci agar pengembang dapat merancang solusi yang tepat (Sopiyan & Darajatun, 2024). Dalam tahap analisis kebutuhan, dilakukan pengumpulan data yang bertujuan untuk menyediakan media interaktif bagi siswa SD, dengan fokus pada materi yang relevan dan menarik. Analisis ini juga mencakup penentuan kebutuhan sistem, seperti platform pengembangan berbasis web, serta fitur yang diterapkan, misalnya desain tampilan yang sesuai untuk anak-anak, efek suara yang mendukung pembelajaran, mekanisme permainan edukatif, dan elemen tambahan lainnya. Tahap ini dirancang untuk

memahami permasalahan yang dihadapi siswa sekaligus merumuskan fungsi, fitur, dan batasan yang diperlukan dalam pengembangan sistem.

## 2. Design (Desain)

Design merupakan tahap dimana pengembang menyusun rencana atau blueprint yang mendetail mengenai bagaimana sistem akan dibangun. Proses ini mencakup berbagai elemen penting, seperti struktur data yang akan diterapkan, arsitektur sistem yang mendasari operasional sistem, serta desain antarmuka pengguna yang akan digunakan. Selain itu, desain juga melibatkan pemilihan algoritma yang diperlukan untuk memastikan kinerja sistem yang optimal (Rahayu & Rahayu, 2017). Tujuan dari desain adalah untuk memberikan gambaran teknis yang jelas tentang komponen-komponen sistem, sehingga proses pengembangan dapat berlangsung lebih terstruktur dan memastikan integrasi antar bagian sistem berjalan dengan lancar.



Gambar 2. Flowchart

<p>Desain esboch di sampul, membuat menu utama dari game berdasarkan tema atau tema ini menaikan halaman menu.</p>	
<p>Desain esboch di sampul, membuat tampilan dari menu yang memiliki 3 level yaitu level 1, level 2, dan level 3.</p>	
<p>Foto esboch di sampul terdapat tampilan dari menu level 1. Pada level 1 itu menggunakan latar belakang hijau dan karakter siobakus.</p>	
<p>Foto esboch di sampul terdapat tampilan dari menu level 2. Pada level 2 itu menggunakan latar belakang hijau dan karakter siobakus.</p>	
<p>Foto esboch di sampul terdapat tampilan dari menu level 3. Pada level 3 itu menggunakan latar belakang hijau dan karakter siobakus.</p>	

Gambar 3. Storyboard

## 3. Implementation (Implementasi)

Pada tahap Implementation, pengembang mulai menulis kode program berdasarkan desain yang telah disusun sebelumnya (Fadli, 2021). Di sini, pengembang menerjemahkan rancangan sistem ke dalam kode yang dapat dieksekusi dengan menggunakan bahasa pemrograman yang sesuai, mewujudkan fungsi dan fitur yang sudah direncanakan dalam desain.

## 4. Verification (Verifikasi)

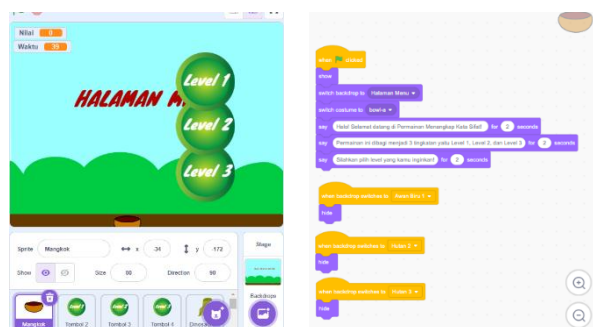
Pada tahap verification, dilakukan proses pengujian untuk memastikan bahwa sistem atau perangkat lunak beroperasi sesuai dengan spesifikasi yang telah ditentukan (Utami dan Yuningsih, 2024). Pengujian ini mencakup berbagai jenis pengujian, seperti pengujian unit, pengujian intergritas, dan pengujian sistem, untuk memastikan bahwa semua fitur berfungsi dengan baik dan sesuai dengan tujuan pembelajaran yang telah dirancang untuk siswa SD.

## 5. Maintenance (Pemeliharaan)

Tahap maintenance dilakukan setelah perangkat lunak atau game dirilis, dengan tujuan untuk menjaga agar sistem berjalan dengan lancar (Rizaldi dkk., 2021). Proses ini mencakup perbaikan kesalahan atau bug, menyesuaikan terhadap perubahan yang terjadi, serta pembaruan fitur dan konten sesuai dengan kebutuhan pengguna.

## HASIL DAN PEMBAHASAN

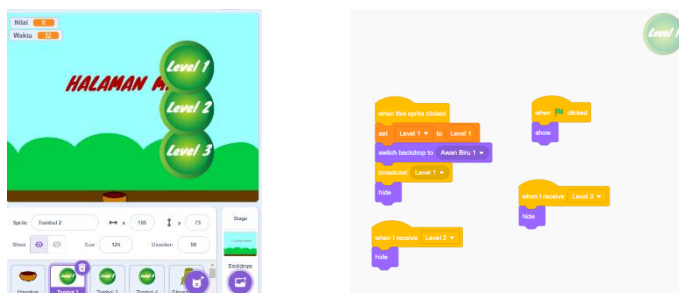
### 1. Sprite Mangkok



Permainan Kata Sifat Berbasis Scratch untuk Melatih Kemampuan Membedakan Kata Sifat di Sekolah Dasar menggunakan sprite berbentuk Mangkok sebagai panduan untuk interaksi awal. Sprite mangkok memiliki blok kode pertama yang dimulai dengan "When green flag clicked" kategori *Events*, yang berfungsi untuk memulai permainan. Setelah itu, perintah "show" digunakan untuk menampilkan sprite Mangkok di layar. Blok "switch backdrop to" kategori *Looks*, mengubah latar belakang menjadi menu utama, sementara "switch costume to" kategori *Looks* mengatur kostum sprite menjadi mangkok dalam posisi awal. Selanjutnya, sprite menggunakan perintah "say" untuk menyampaikan pesan kepada pemain, yaitu: "Halo" selama 2 detik, "Permainan ini memiliki 3 tingkatan yaitu level 1, level 2, dan level 3" selama 2 detik, dan "Silakan pilih level" selama 2 detik.

Blok "when green flag clicked" yang berada dalam kategori *Events* berfungsi untuk memulai permainan, sedangkan blok "show" dalam kategori *Looks* digunakan untuk menampilkan sprite di layar. Blok "switch backdrop to" dan "switch costume to" juga terletak dalam kategori *Looks*, yang bertugas mengubah tampilan latar belakang dan kostum sprite. Perintah "say" digunakan untuk menampilkan pesan teks sesuai durasi yang ditentukan dalam permainan

## 2. Tombol 2



“when this sprite clicked” kategori *Events* berfungsi memulai rangkaian perintah ketika sprite ini diklik oleh pengguna, “set Level 1 to Level 1” kategori *Variables* berfungsi mengatur variables Level 1 dengan nilai kemungkinan sudah didefinisikan sebelumnya sebagai Level 1. Biasanya ini digunakan untuk mengatur status permainan atau level saat ini, “switch backdrop to Awan Biru 1” kategori *Looks* berfungsi mengganti latar backdrop ke Awan Biru 1. Latar ini mungkin menggambarkan perubahan suasana dalam permainan atau cerita, “broadcast Level 1” kategori *Events* berfungsi mengirim pesan siaran dengan nama Level 1 ke semua sprite dan skrip dalam proyek. Pesan ini bisa digunakan untuk memicu aksi lain di sprite yang berbeda, “hide” kategori *Looks* berfungsi menyembunyikan sprite ini dari tampilan layar. Kode ini mungkin digunakan dalam permainan atau cerita interaktif untuk memulai sebuah level baru ketika sprite diklik. Gambar tersebut menunjukkan kode pemrograman berbasis blok dari platform seperti Scratch.

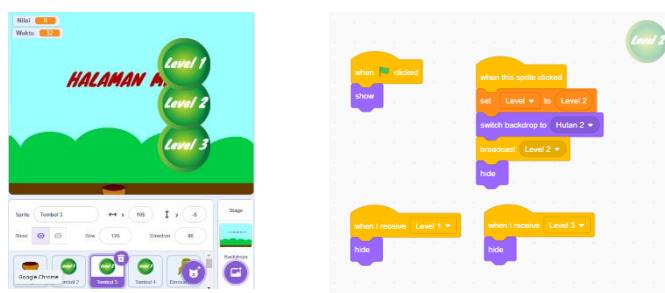
“when green flag clicked” kategori *Events* berfungsi blok ini adalah event handler yang memulai skrip ketika bendera hijau (ikon "start") di klik. Ini adalah pemicu utama untuk menjalankan kode, “show” kategori *Looks* berfungsi blok ini membuat sprite (objek atau karakter dalam proyek) terlihat di panggung. Jika sprite dalam keadaan tersembunyi sebelumnya, kode ini akan membuatnya muncul.

“when I receive Level 2” kategori *Events* berfungsi blok ini memulai skrip ketika pesan "Level 2" diterima. Pesan ini biasanya dikirim oleh blok "broadcast" di program Scratch. Hal ini digunakan untuk mengoordinasikan aksi antar sprite atau antar bagian skrip, “hide” kategori *Looks* berfungsi blok ini menyembunyikan sprite dari panggung. Sprite tidak akan terlihat meskipun masih ada di program dan dapat muncul kembali dengan blok "show".

“when I receive level 3” kategori *Events* memicu aksi tertentu saat menerima pesan "level 3", seperti memulai level baru atau mengubah tampilan, “hide” kategori *Looks* menyembunyikan sprite dari layar. Biasanya digunakan untuk objek yang tidak diperlukan di level tertentu.



### 3. Tombol 3



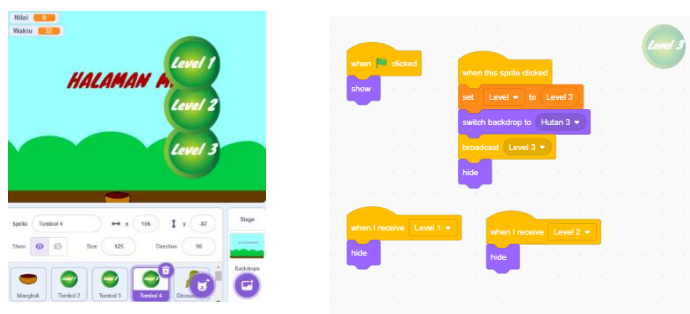
“when this sprite clicked” kategori *Events* berfungsi untuk memulai rangkaian perintah ketika sprite ini diklik oleh pengguna, “set Level to Level 2” kategori *Variables* berfungsi untuk mengatur variabel Level 2 dengan nilai yang kemungkinan sudah didefinisikan sebelumnya sebagai Level 2. Biasanya, kategori ini digunakan untuk mengatur status permainan atau level yang sedang berlangsung, “switch backdrop to Hutan 2” kategori *Looks* berfungsi untuk mengganti latar belakang dengan Hutan 2. Latar belakang ini mungkin menggambarkan perubahan suasana dalam permainan atau cerita, “broadcast Level 2” kategori *Events* berfungsi untuk mengirimkan pesan siaran dengan nama Level 2 kepada semua sprite dan skrip dalam proyek. Pesan ini dapat digunakan untuk memicu aksi-aksi lain pada sprite yang berbeda, “hide” kategori *Looks* berfungsi untuk menyembunyikan sprite ini dari tampilan layar. Kode ini kemungkinan digunakan dalam permainan atau cerita interaktif untuk memulai level baru ketika sprite diklik. Gambar tersebut menunjukkan kode pemrograman berbasis blok dari platform seperti Scratch.

“when green flag clicked” kategori *Events* berfungsi blok ini sebagai event handler yang memulai skrip ketika bendera hijau (ikon “start”) diklik. Ini adalah pemicu utama untuk menjalankan kode, “show” kategori berfungsi blok ini untuk membuat sprite (objek atau karakter dalam proyek) terlihat di panggung. Jika sprite sebelumnya disembunyikan, kode ini akan membuatnya muncul kembali.

“when I receive Level 1” kategori *Events* berfungsi blok ini untuk memulai skrip saat pesan “Level 1” diterima. Pesan ini biasanya dikirim oleh blok “broadcast” dalam program Scratch. Fungsi ini digunakan untuk mengoordinasikan aksi antar sprite atau antar bagian skrip, “hide” kategori *Looks* berfungsi blok ini untuk menyembunyikan sprite dari panggung. Sprite tersebut tidak akan terlihat meskipun tetap ada dalam program dan bisa muncul lagi dengan blok “show”.

“when I receive level 3” kategori *Events* ini memicu aksi tertentu saat menerima pesan “level 3”, seperti memulai level baru atau mengubah tampilan, “hide” kategori *Looks* berfungsi untuk menyembunyikan sprite dari layar. Biasanya, ini digunakan untuk objek yang tidak diperlukan pada level tertentu.

#### 4. Tombol 4



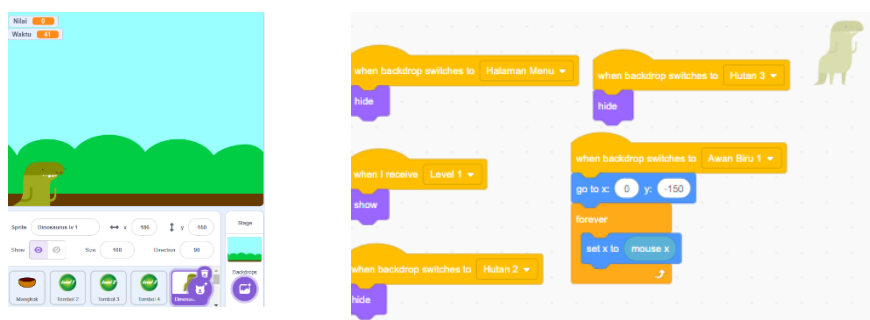
“when this sprite clicked” kategori *Events* berfungsi untuk memulai rangkaian perintah ketika sprite ini diklik oleh pengguna, “set Level to Level 3” kategori *Variables* berfungsi untuk mengatur variabel Level 3 dengan nilai yang kemungkinan sudah didefinisikan sebelumnya sebagai Level 3. Biasanya, kategori ini digunakan untuk mengatur status permainan atau level yang sedang berlangsung, “switch backdrop to Hutan 3” kategori *Looks* berfungsi untuk mengganti latar belakang dengan Hutan 3. Latar belakang ini mungkin menggambarkan perubahan suasana dalam permainan atau cerita, “broadcast Level 3” kategori *Events* berfungsi untuk mengirimkan pesan siaran dengan nama Level 3 kepada semua sprite dan skrip dalam proyek. Pesan ini dapat digunakan untuk memicu aksi-aksi lain pada sprite yang berbeda, “hide” kategori *Looks* berfungsi untuk menyembunyikan sprite ini dari tampilan layar. Kode ini kemungkinan digunakan dalam permainan atau cerita interaktif untuk memulai level baru ketika sprite diklik. Gambar tersebut menunjukkan kode pemrograman berbasis blok dari platform seperti Scratch.

“when green flag clicked” kategori *Events* berfungsi sebagai event handler yang memulai skrip ketika bendera hijau (ikon “start”) diklik. Ini adalah pemicu utama untuk menjalankan kode, “show” kategori *Looks* berfungsi untuk membuat sprite (objek atau karakter dalam proyek) terlihat di panggung. Jika sprite sebelumnya disembunyikan, kode ini akan membuatnya muncul kembali.

“when I receive Level 1” kategori *Events* berfungsi untuk memulai skrip saat pesan “Level 1” diterima. Pesan ini biasanya dikirim oleh blok “broadcast” dalam program Scratch. Fungsi ini digunakan untuk mengoordinasikan aksi antar sprite atau antar bagian skrip, “hide” kategori *Looks* berfungsi untuk menyembunyikan sprite dari panggung. Sprite tersebut tidak akan terlihat meskipun tetap ada dalam program dan bisa muncul lagi dengan blok “show”.

“when I receive level 2” kategori *Events* ini memicu aksi tertentu saat menerima pesan “level 2”, seperti memulai level baru atau mengubah tampilan, “hide” kategori *Looks* berfungsi untuk menyembunyikan sprite dari layar. Biasanya, ini digunakan untuk objek yang tidak diperlukan pada level tertentu.

## 5. Dinosaurus karakter utama pada level 1



"when backdrop switches to [halaman menu]" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi halaman menu biasanya digunakan untuk mengatur elemen yang hanya muncul di menu, seperti tombol mulai atau logo, "hide" kategori *Looks* berfungsi menyembunyikan sprite dari layar digunakan untuk sprite yang tidak relevan di halaman menu, seperti karakter atau elemen dari level permainan.

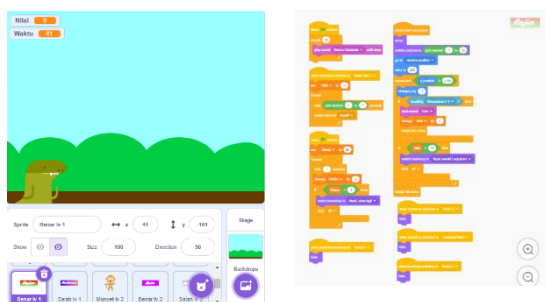
"when I receive [level 1]" kategori *Events* berfungsi memicu skrip tertentu saat pesan "level 1" diterima digunakan untuk memulai aksi atau menampilkan elemen yang relevan di level 1, seperti karakter, objek, atau latar, "show" kategori *Looks* berfungsi menampilkan sprite di layar digunakan untuk memastikan sprite yang diperlukan di level 1 terlihat.

"when backdrop switches to [hutan 2]" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 2. digunakan untuk mengatur elemen atau aksi yang hanya terjadi di latar hutan 2, "hide" kategori *Looks* berfungsi menyembunyikan sprite dari layar digunakan untuk sprite yang tidak diperlukan saat latar berada di hutan 2, seperti elemen dari level sebelumnya atau menu.

"when backdrop switches to [hutan 3]" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 3 biasanya untuk mengatur elemen atau aksi yang relevan dengan latar hutan 3, seperti memulai animasi, menampilkan elemen baru, atau mengatur ulang posisi sprite, "hide" kategori *Looks* berfungsi menyembunyikan sprite dari layar untuk menyembunyikan sprite yang tidak diperlukan di latar hutan 3, seperti elemen dari latar sebelumnya.

"when backdrop switches to [awan biru 1]" kategori *Events* berfungsi memicu skrip saat latar belakang berubah menjadi awan biru 1 digunakan untuk memulai aksi yang relevan saat latar ini aktif, misalnya mengatur posisi sprite, "go to x: 0 y: -150" Kategori *Motion* berfungsi memindahkan sprite ke posisi koordinat x: 0, y: -150 pada layar untuk menempatkan sprite pada posisi tertentu di awal permainan atau level, "forever set x to mouse x" kategori *Control* dan *Motion* berfungsi dalam loop forever, sprite akan terus mengikuti posisi x dari kursor mouse biasanya digunakan untuk kontrol pemain (misalnya karakter atau paddle) yang bergerak mengikuti arah gerakan mouse di sepanjang sumbu x.

## 6. Benar level 1



“When greenflag clicked” kategori *Events* ini adalah blok yang mengindikasikan bahwa perintah di bawahnya akan dijalankan ketika pengguna mengklik bendera hijau (green flag). Bendera hijau sering digunakan untuk memulai sebuah program di Scratch, “repeat 10” kategori *Control* blok ini berarti perintah yang ada di dalamnya akan dijalankan sebanyak 10 kali. Jadi, semua perintah setelahnya akan diulang sebanyak 10 kali, “play sound dance celebrate until done” kategori *Sound* Ini adalah blok untuk memainkan suara "dance celebrate". Kata "until done" berarti suara akan dimainkan hingga selesai tanpa gangguan. Jadi, secara keseluruhan, kode ini akan menjalankan suara "dance celebrate" sebanyak 10 kali setiap kali bendera hijau diklik.

“When backdrop switches to awan biru 1” kategori *Events* perintah ini berarti program akan mulai mengeksekusi blok-blok berikutnya hanya ketika latar belakang (backdrop) berubah menjadi "awan biru 1". Jadi, blok berikutnya akan dijalankan jika latar belakang proyek diubah ke "awan biru 1", “set nilai to 0” kategori *Variables* blok ini akan mengatur nilai dari *Variables* yang bernama "nilai" menjadi 0. Jadi, setiap kali perintah ini dijalankan, nilai *variables* "nilai" akan diset menjadi 0, “Forever” kategori *Control* blok ini memastikan bahwa perintah di dalamnya akan terus dijalankan tanpa henti, selama program berjalan, “wait pick random 1 to 1 seconds” kategori *Operators* perintah ini membuat sprite berhenti sejenak selama waktu acak antara 1 hingga 1 detik. Karena angka awal dan akhir adalah sama (1 detik), maka sprite akan menunggu selama 1 detik sebelum melanjutkan ke perintah berikutnya, “create clone of myself” kategori *Control* perintah ini membuat salinan (clone) dari sprite yang menjalankan kode tersebut. Jadi, sprite akan menggandakan dirinya sendiri setiap kali perintah ini dijalankan.

“When greenflag clicked” kategori *Events* perintah ini berarti blok-blok berikutnya akan dijalankan ketika pengguna mengklik bendera hijau. Biasanya digunakan untuk memulai program di Scratch, “set waktu to 50” terletak pada kode *Variables* blok ini akan mengatur nilai variabel "waktu" menjadi 50. Jadi, *variables* "waktu" dimulai dengan nilai 50 saat bendera hijau diklik, “Forever” kategori *Control* blok ini memastikan bahwa perintah yang ada di dalamnya akan terus dijalankan tanpa henti, selama program berjalan, “wait 1 seconds” kategori *Control* perintah ini akan membuat sprite berhenti sejenak selama 1 detik setiap kali dijalankan. Ini berarti

sprite akan menunggu 1 detik sebelum melanjutkan perintah berikutnya, "change waktu by -1" terletak pada kode *Variables* perintah ini akan mengurangi nilai variabel "waktu" sebanyak 1 setiap kali dijalankan. Jadi, setiap 1 detik (sesuai dengan blok "wait"), nilai "waktu" akan berkurang sebesar 1, "If waktu < 1 then" kategori *Control*, *Operators*, dan *Variables* ini adalah kondisi yang memeriksa apakah nilai variabel "waktu" kurang dari 1. Jika ya, maka perintah di dalam blok "if" akan dijalankan, "Switch backdrop to maaf, coba lagi!" kategori *Looks* jika kondisi "waktu < 1" terpenuhi, latar belakang proyek akan berubah menjadi "maaf, coba lagi!". Ini memberi tahu pemain atau pengguna bahwa waktu telah habis, "stop all" kategori *Control* setelah latar belakang berubah, program akan berhenti sepenuhnya. Semua sprite dan perintah lainnya akan dihentikan.

"when backdrop switches to hutan 2" kategori *Events* berfungsi memicu skrip saat latar belakang berubah menjadi hutan 2. Biasanya digunakan untuk memulai aksi atau perubahan tampilan ketika latar hutan 2 muncul, seperti menyembunyikan atau menampilkan elemen tertentu, hide terletak pada kode *Looks* berfungsi menyembunyikan sprite dari layar di gunakan untuk menyembunyikan sprite yang tidak diperlukan lagi saat berada di latar hutan 2, seperti elemen dari level sebelumnya atau objek yang tidak relevan dengan latar baru.

"When I start as a clone" kategori *Control* blok ini berarti perintah di bawahnya akan dijalankan setiap kali sprite ini dibuat sebagai clone (salinan). Jadi, ketika sprite digandakan, perintah berikutnya akan dijalankan, "Show" kategori *Looks* ini memastikan bahwa clone akan muncul di layar setelah dibuat, "switch costume to pick random 1 to 10" kategori *Looks* blok ini akan memilih kostum (penampilan) acak dari kostum yang ada, dengan memilih angka acak antara 1 hingga 10. Jadi, setiap clone yang dibuat akan memiliki kostum yang berbeda secara acak dari daftar yang ada, "Go to random position" kategori *Motion* blok ini akan menggerakkan clone ke posisi acak di layar. Setiap clone yang muncul akan berada di tempat yang berbeda-beda, "set y to 500" kategori *Motion* blok ini mengatur posisi vertikal clone (sumbu Y) ke 500. Jadi, clone akan ditempatkan pada posisi tinggi di layar pada sumbu Y, "Repeat until y position kurang dari -150" kategori *Control*, *Operators* dan *Motion* blok ini berarti perintah di dalamnya akan dijalankan berulang-ulang sampai posisi Y clone kurang dari -150. Selama posisi Y lebih besar dari -150, perintah di dalam loop akan terus dijalankan, "change y by -1" terletak pada kode *Motion* setiap kali loop dijalankan, posisi Y clone akan berkurang sebesar 1. Dengan demikian, clone akan bergerak ke bawah secara perlahan, dari  $Y = 500$  hingga  $Y < -150$ , "If touching dinosaur lv 1?" If Kategori *Control* touching kategori *Sensing* setelah posisi Y clone turun, kondisi ini memeriksa apakah clone sedang menyentuh sprite bernama "dinosaur lv 1." Jika ya, perintah dalam blok "if" akan dijalankan, "start sound coin" kategori *Sound* Jika clone menyentuh dinosaur, suara "coin" akan diputar, menandakan bahwa ada interaksi yang berhasil antara clone dan dinosaur, "change nilai by 1" kategori *Variables* setiap kali clone menyentuh dinosaur, nilai variabel "nilai" akan bertambah 1, "Delete this clone" kategori *Control* setelah clone

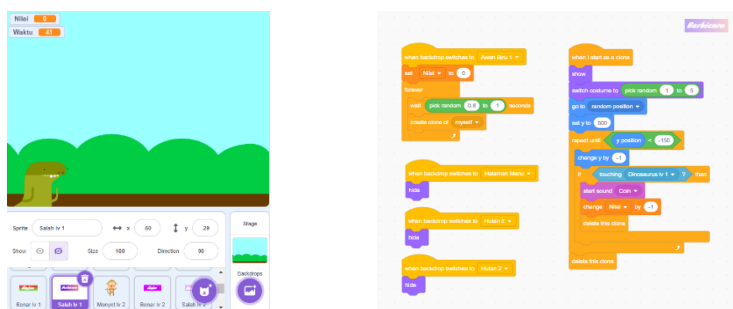
berinteraksi dengan dinosaurus dan nilai bertambah, clone tersebut akan dihapus dari layar, menghilangkan salinan sprite yang telah berfungsi, "If nilai lebih dari 10 then" kategori *Control*, *Operators*, dan *Variables* blok ini memeriksa apakah nilai sudah lebih besar dari 10. Jika ya, perintah di dalam blok "if" akan dijalankan, "switch backdrop to tepat sekali! Lanjutkan!" kategori *Looks*, Jika nilai lebih dari 10, latar belakang akan berubah menjadi "tepat sekali! Lanjutkan!" untuk memberi tahu pemain bahwa mereka telah mencapai tujuan atau pencapaian tertentu, "stop all" kategori *Control* setelah latar belakang berubah, perintah ini menghentikan semua perintah atau sprite yang berjalan, menyelesaikan permainan atau level ini, "Delete this clone" kategori *Control*, setelah proses selesai, clone yang sedang berinteraksi dihapus.

"when backdrop switches to hutan 2" kategori *Events* blok ini berarti perintah di bawahnya akan dijalankan ketika latar belakang (backdrop) proyek berubah menjadi "hutan 2." Jadi, perintah berikutnya hanya akan diaktifkan jika latar belakang yang ditampilkan adalah "hutan 2", "hide" kategori *Looks*, blok ini menyembunyikan sprite yang menjalankan perintah tersebut. Jadi, saat latar belakang berubah menjadi "hutan 2," sprite akan disembunyikan dari layar.

"when backdrop switches to halaman menu" terletak pada kode *Events* blok ini berarti perintah di bawahnya akan dijalankan ketika latar belakang (backdrop) proyek berubah menjadi "halaman menu." Perintah ini hanya akan aktif jika latar belakang yang ditampilkan adalah "halaman menu", "Hide" kategori *Looks*, blok ini menyembunyikan sprite yang menjalankan perintah ini. Jadi, saat latar belakang berubah menjadi "halaman menu," sprite tersebut akan disembunyikan dari layar.

"when backdrop switches to hutan 3" terletak pada kode *Events* blok ini berarti perintah di bawahnya akan dijalankan ketika latar belakang (backdrop) proyek berubah menjadi "hutan 3." perintah ini hanya akan aktif jika latar belakang yang ditampilkan adalah "hutan 3" "Hide" kategori *Looks*, blok ini menyembunyikan sprite yang menjalankan perintah ini. Jadi, ketika latar belakang berubah menjadi "hutan 3," sprite tersebut akan disembunyikan dari layar.

## 7. Salah level 1



"when backdrop switches to Awan Biru" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "Awan Biru." Jadi, kode berikutnya akan dijalankan hanya ketika latar belakang berada dalam kondisi tersebut, "set nilai to 0" kategori *Variables* setelah latar belakang berubah, kode ini akan mengatur nilai *Variables* bernama "nilai" menjadi, "forever" kategori *Control* kode ini berarti bagian berikutnya akan dijalankan terus-menerus tanpa henti, selama kondisi sebelumnya terpenuhi, "wait pick random 0.8 to 1 seconds" kategori *Events* dan *Operators* setelah itu, program akan berhenti sejenak (menunggu) selama waktu acak antara 0.8 hingga 1 detik. Ini menciptakan jeda yang bervariasi antara setiap aksi yang dilakukan, "create clone of myself" kategori *Control* setelah menunggu, program akan membuat salinan (clone) dari sprite yang menjalankan kode ini.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu, "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when backdrop switches to hutan 2" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "hutan 2." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "hutan 2", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi. Secara keseluruhan, kode ini akan menyembunyikan sprite setiap kali latar belakang berubah menjadi "hutan 2."

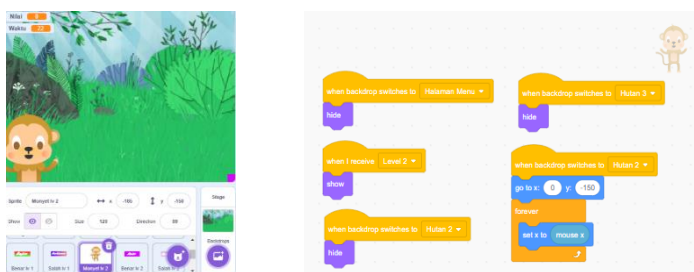
"when backdrop switches to hutan 2" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "hutan 2." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "hutan 2", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi. Secara keseluruhan, kode ini akan menyembunyikan sprite setiap kali latar belakang berubah menjadi "hutan 2."

"when I start as a clone" kategori *Control* kode ini diaktifkan saat sebuah clone dari sprite ini dibuat, "show" kategori begitu clone dibuat, sprite akan muncul (tampil di layar), "switch costume to pick random 1 to 5" kategori *Looks* dan *Operators* clone akan memilih salah satu kostum secara acak dari nomor 1 hingga 5, kostum ini biasanya digunakan untuk mengganti tampilan atau penampilan sprite, "go to random position" kategori *Control* klon akan diposisikan secara acak di layar, "set y

to 500" kategori *Motion* setelah posisi acak dipilih, posisi vertikal (sumbu y) klon akan diatur menjadi 500, sehingga klon akan muncul di bagian atas layar, "repeat until y position < -150" kategori *Control* ini akan menjalankan blok di dalamnya berulang-ulang sampai posisi y clone turun di bawah -150 (di luar layar bawah), "change y by -1" kategori selama perulangan, posisi y clone akan terus berkurang sebanyak 1 tiap kali perulangan, membuatnya bergerak ke bawah, "if touching dinosaur lv 1?" kategori *Control* dan *Sensing* jika clone menyentuh objek bernama "dinosaurus level 1," maka aksi berikutnya akan dijalankan, "start sound coin" kategori begitu clone menyentuh dinosaurus, suara "coin" akan dimainkan (biasanya suara koin yang terkumpul, "change nilai by -1" kategori *Motion* nilai *Variables* "nilai" akan berkurang sebanyak 1 setiap kali clone menyentuh dinosaurus level1, "delete this clone" kategori *Control*.

setelah itu, klon akan dihapus (dihapus dari layar). Secara keseluruhan, kode ini mengatur bahwa ketika sebuah clone muncul, ia akan bergerak ke bawah dan jika menyentuh objek tertentu ("dinosaurus level 1"), maka akan memainkan suara dan mengurangi nilai. Setelah itu, clone akan dihapus.

## 8. Monyet karakter utama pada level 2



"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu," "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when I receive level 2" kategori *Events* berfungsi memicu skrip tertentu saat pesan "level 2" diterima, digunakan untuk memulai aksi atau menampilkan elemen yang relevan di level 2, seperti karakter, objek, atau latar, "show" kategori *Looks* berfungsi menampilkan sprite layar digunakan untuk memastikan sprite yang diperlukan di level 2 terlihat.

"when backdrop switches to hutan 2" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 2. digunakan untuk mengatur elemen atau aksi yang hanya terjadi di latar hutan 2, "hide" kategori *Looks* berfungsi

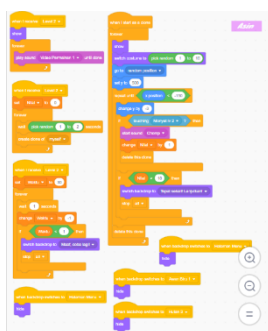
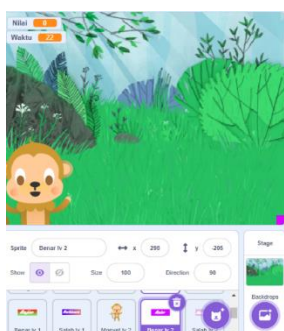


menyembunyikan sprite dari layar, digunakan untuk sprite yang tidak diperlukan saat latar berada di hutan 2, seperti elemen dari level sebelumnya atau menu.

"when backdrop switches to hutan 3" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 3 biasanya untuk mengatur elemen atau aksi yang relevan dengan latar hutan 3, seperti memulai animasi, menampilkan elemen baru, atau mengatur ulang posisi sprite, "hide" kategori *Looks* berfungsi menyembunyikan untuk menyembunyikan sprite yang tidak diperlukan di latar hutan 3, seperti elemen dari latar sebelumnya.

"when backdrop switches to hutan 2" kategori *Events* berfungsi memicu skrip ketika latar belakang berubah menjadi hutan 2, digunakan untuk memulai aksi yang relevan saat latar ini aktif, seperti memindahkan sprite ke posisi tertentu, "go to x: 0 y: -150" kategori *Motion* berfungsi memindahkan sprite ke posisi koordinat x: 0, y: -150 berfungsi mengatur posisi sprite pada koordinat tersebut ketika latar berubah menjadi hutan 2, biasanya untuk memulai posisi yang konsisten, "forever" kategori *Control* menjalankan skrip yang ada di dalamnya secara terus-menerus tanpa henti digunakan untuk menjalankan perintah yang harus tetap aktif selama permainan berlangsung, seperti mengikuti pergerakan mouse, "set x to mouse x" kategori *Motion* untuk menetapkan posisi x sprite agar selalu mengikuti posisi x kursor mouse. membuat sprite bergerak mengikuti pergerakan mouse di sepanjang sumbu x, biasanya digunakan untuk kontrol pemain seperti karakter atau objek yang dikendalikan dengan mouse.

## 9. Benar level 2



"when I receive level 2" kategori *Events* kode ini akan dijalankan ketika sprite menerima pesan dengan nama "level 2." Pesan ini biasanya dikirim dari bagian lain dari program, dan begitu pesan diterima, kode berikutnya akan aktif, "show" kategori *Looks* setelah menerima pesan "level 2," sprite akan muncul di layar jika sebelumnya disembunyikan. Jadi, bagian ini memastikan sprite terlihat setelah menerima pesan tersebut, "forever" kategori *Control* bagian ini menunjukkan bahwa aksi berikutnya akan dijalankan terus-menerus, selama kondisi sebelumnya (menerima pesan "level 2") terpenuhi, "play sound wideo permainan 1 until done" kategori *Sound* di dalam perulangan forever, suara "video permainan 1" akan diputar, dan kode ini akan

menunggu sampai suara selesai diputar sebelum melanjutkan. "until done" memastikan suara akan diputar sampai selesai sebelum kode melanjutkan langkah berikutnya.

"when I receive level 2" kategori *Events* kode ini akan dijalankan ketika sprite menerima pesan dengan nama "level 2." pesan ini biasanya dikirim dari bagian lain dalam program. Ketika sprite menerima pesan ini, kode berikutnya akan diaktifkan, "set nilai to 0" kategori *Variables* setelah menerima pesan "level 2," nilai dari variabel "nilai" akan diatur menjadi 0. Ini biasanya digunakan untuk memulai ulang atau mereset nilai yang digunakan dalam permainan, "forever" kategori *Control* bagian ini menunjukkan bahwa aksi berikutnya akan dijalankan terus-menerus tanpa henti selama kondisi sebelumnya (menerima pesan "level 2") terpenuhi, "wait pick random 1 to 2 seconds" kategori *Control* dan *Operators* setelah itu, program akan berhenti sementara (menunggu) selama waktu acak antara 1 hingga 2 detik. Ini menciptakan jeda acak antara setiap aksi yang dilakukan, "create clone of my self" kategori *Control* setelah menunggu, program akan membuat salinan (clone) dari sprite yang menjalankan kode ini. Setiap kali interval waktu acak selesai, clone baru akan dibuat.

"when I receive level 2" kategori *Events* kode ini diaktifkan ketika sprite menerima pesan dengan nama "level 2." pesan ini biasanya dikirim dari bagian lain dari program. Setelah menerima pesan ini, kode berikutnya akan dijalankan, "set waktu to 30" kategori *Variables* setelah menerima pesan "level 2," nilai dari *Variables* "waktu" akan diatur ke 30. Ini berarti waktu hitung mundur dimulai dari 30 detik, "forever" kategori *Control* bagian ini menunjukkan bahwa aksi berikutnya akan dijalankan terus-menerus selama kondisi sebelumnya (menerima pesan "level 2") tetap terpenuhi, "wait 1 seconds" kategori *Control* di dalam perulangan forever, program akan menunggu selama 1 detik sebelum melanjutkan ke aksi berikutnya. Ini membuat program mengurangi waktu secara bertahap, 1 detik setiap kali, "change waktu by -1" kategori *Variables* setelah menunggu 1 detik, *Variables* "waktu" akan dikurangi sebanyak 1. Ini berarti waktu hitung mundur berkurang 1 detik setiap kali, "if waktu < 1 then" kategori *Control*, *Operators*, dan *Variables* setelah mengurangi waktu, kode akan memeriksa apakah waktu sudah kurang dari 1 detik. Jika waktu menjadi kurang dari 1, maka blok kode berikutnya akan dijalankan, "switch backdrop to maaf, coba lagi!" Kategori *Looks* jika waktu kurang dari 1, latar belakang (backdrop) akan berubah menjadi "maaf, coba lagi!" Ini biasanya menunjukkan bahwa waktu telah habis, dan pemain perlu mencoba lagi, "stop all" kategori *Control* setelah latar belakang berubah, "stop all" akan menghentikan semua skrip yang sedang berjalan. Ini berarti permainan atau aktivitas yang sedang berlangsung akan berhenti.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini

biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when I start as a clone" kategori *Control* kode ini akan dijalankan ketika sprite dibuat sebagai clone (salinan), "forever" kategori *Control* semua aksi di dalam blok ini akan dijalankan terus-menerus sampai clone dihapus atau skrip dihentikan, Langkah 2: Menampilkan dan Mengatur Clone "show" *Looks* kategori Clone akan ditampilkan (jika sebelumnya disembunyikan), "switch costume to pick random 1 to 10" kategori *Looks* dan *Operators* Clone akan memilih tampilan (kostum) secara acak dari nomor 1 hingga 10. Ini digunakan untuk memberikan variasi pada tampilan clone, "go to random position" kategori Clone akan diposisikan secara acak di layar, "set y to 500" kategori *Motion* Clone akan diatur ke posisi vertikal  $y = 500$ , yaitu di luar bagian atas layar. Hal ini sering dilakukan untuk memulai pergerakan dari atas layar ke bawah. Langkah 3: Pergerakan Clone, "repeat until x position  $< -150$ " kategori *Control*, *Sensing*, dan *Operators* Clone akan terus menjalankan blok berikutnya sampai posisi x menjadi kurang dari -150 (di luar layar kiri), "change y by -3" kategori *Motion* setiap pengulangan, posisi y akan dikurangi sebanyak 3, sehingga clone bergerak turun ke bawah layar. Langkah 4: Interaksi dengan "Monyet level 2", "if touching monyet lv 2?" kategori *Control* dan *Sensing* jika clone menyentuh sprite "monyet level 2," maka aksi berikutnya akan dijalankan, "start sound chomp" kategori suara "chomp" akan dimainkan untuk memberikan efek audio ketika clone berinteraksi dengan "monyet lv 2.", "change nilai by 1" kategori *Variables* "nilai" akan bertambah sebanyak 1 setiap kali clone menyentuh "monyet lv 2.", "delete this clone" kategori Clone akan dihapus setelah menyentuh "monyet level 2. Langkah 5: Cek Kondisi Kemenangan, "if nilai  $> 10$  then" kategori jika nilai *Variables* "nilai" lebih besar dari 10, maka aksi berikutnya akan dijalankan, "switch backdrop to tepat sekali! Lanjutkan!" kategori Latar belakang (backdrop) akan berubah menjadi "tepat sekali! Lanjutkan!" untuk menunjukkan bahwa pemain telah mencapai target, "stop all" kategori *Control* semua skrip dalam proyek akan dihentikan. Ini biasanya menunjukkan akhir level atau permainan, "delete this clone" kategori Clone akan dihapus setelah kondisi kemenangan tercapai.

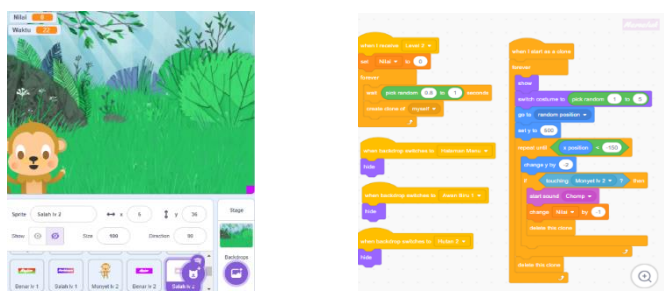
"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu, "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when backdrop switches to awan biru 1" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "awan biru 1." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "awan biru 1, "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini

berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

"when backdrop switches to "hutan 3" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 3 biasanya untuk mengatur elemen atau aksi yang relevan dengan latar hutan 3, seperti memulai animasi, menampilkan elemen baru, atau mengatur ulang posisi sprite, "hide" kategori *Looks* berfungsi menyembunyikan sprite dari layar untuk menyembunyikan sprite yang tidak diperlukan di latar hutan 3, seperti elemen dari latar sebelumnya.

## 10. Salah level 2



"When I receive level 2" kategori *Events* blok ini menandakan bahwa skrip akan dijalankan ketika pesan "level 2" diterima. Pesan ini biasanya dikirimkan oleh skrip lain untuk menandakan bahwa program telah mencapai level tertentu, set nilai to blok ini mengatur Variables "nilai" menjadi 0. Variables ini mungkin digunakan untuk menghitung skor, jumlah klon, atau statistik lain yang relevan dengan permainan, "Forever" kategori *Control* blok ini membuat perintah di dalamnya berjalan terus-menerus tanpa berhenti, selama program masih aktif "Wait [pick random 0.8 to 1] seconds" kategori *Control* dan *Operators* perintah ini memberi jeda secara acak antara 0.8 hingga 1 detik sebelum melanjutkan ke langkah berikutnya. Hal ini memberikan variasi pada waktu eksekusi, sehingga klon tidak muncul pada interval waktu yang sama, "Create clone of [myself]" kategori *Control* blok ini membuat klon dari objek yang menjalankan skrip ini. Clone adalah salinan dari objek yang bisa memiliki perilaku independen.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu", "hide" kategori *Looks* setelah etelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

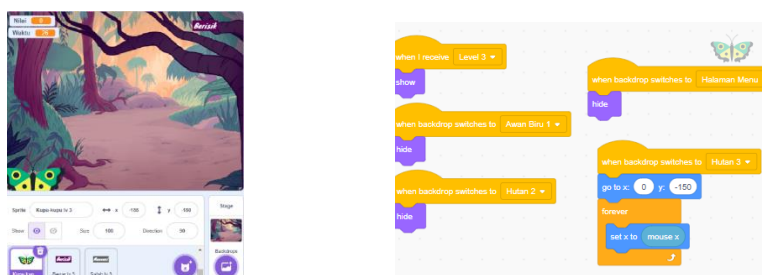
"when backdrop switches to awan biru 1" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi

"awan biru 1." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "awan biru 1", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

"when backdrop switches to hutan 2" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "hutan 2." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "hutan 2", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

"when I start as a clone" kategori *Control* blok ini menandakan bahwa skrip akan dijalankan ketika objek ini menjadi klon. Setiap klon yang dibuat akan mengikuti instruksi yang ada di dalam blok ini, "Forever" kategori *Control* blok ini memastikan bahwa instruksi di dalamnya berjalan terus-menerus selama klon tersebut tidak dihapus, "show" kategori *Looks* blok ini menampilkan klon pada layar, memastikan bahwa klon yang baru dibuat terlihat, "switch costume to [pick random 1 to 5]" kategori *Looks* dan *Operators* blok ini mengubah kostum objek secara acak ke salah satu dari lima kostum yang tersedia. Ini digunakan untuk memberikan variasi tampilan pada klon, misalnya agar terlihat berbeda setiap kali muncul, "go to [random position]" kategori *Motion* Clone diposisikan secara acak di layar, menciptakan elemen kejutan, "set y to 500" kategori *Motion* Clone dipindahkan ke koordinat vertikal  $y = 500$ , biasanya berada di luar layar bagian atas, "Repeat until [x position < -150]" kategori *Control*, *Operators* dan *Sensing* perulangan ini membuat klon bergerak ke bawah sampai posisi  $x$  klon lebih kecil dari -150, biasanya untuk memastikan klon bergerak melewati area tertentu, "Change y by -2" kategori *Motion* selama perulangan, posisi vertikal  $y$  klon akan berkurang sebesar 2 setiap langkah, sehingga klon terlihat bergerak ke bawah layar, "If [touching monyet lv 2?] Then" kategori *Control* dan *Sensing* mengecek apakah klon menyentuh objek bernama monyet level 2. Jika iya, perintah di dalam kondisi ini akan dijalankan, "start sound [chomp]" kategori *Sound* memainkan efek suara "chomp", biasanya untuk memberikan umpan balik auditori bahwa interaksi terjadi, "Change [nilai] by -1" kategori *Variables* mengurangi nilai variabel "nilai" sebesar 1. Ini mungkin digunakan untuk menandai kehilangan skor atau nyawa pemain saat terjadi kontak dengan Clone, "delete this clone" kategori *Control* menghapus clone dari permainan setelah interaksi selesai, sehingga klon tidak terus muncul di layar, "delete this clone (diulang)" kategori meskipun disebut dua kali, fungsi ini hanya perlu satu kali karena klon akan dihapus begitu perintah ini dijalankan.

## 11. Kupu-kupu karakter utama pada level 3



"when I receive level 3" kategori *Events* kode ini akan dijalankan ketika sprite menerima pesan dengan nama "level 3." Pesan ini biasanya dikirim dari bagian lain dalam program untuk menandai bahwa permainan telah memasuki level 3, "show" kategori *Looks* setelah menerima pesan "level 3," sprite akan ditampilkan di layar. Jika sprite sebelumnya disembunyikan (dengan hide), maka perintah show ini memastikan sprite muncul kembali.

"when backdrop switches to awan biru 1" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "awan biru 1." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "awan biru 1", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

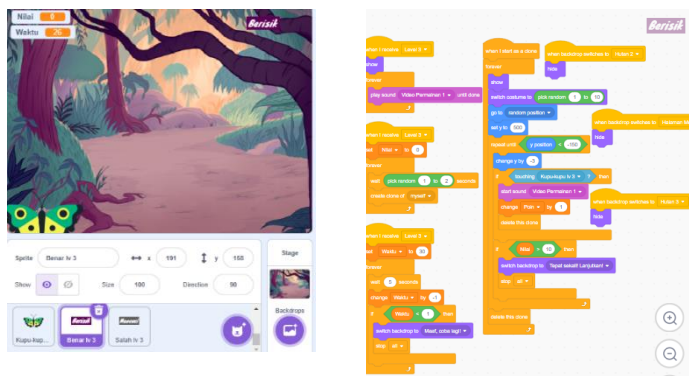
"when backdrop switches to [hutan 2]" *Events* kategori berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 2 digunakan untuk mengatur elemen atau aksi yang hanya terjadi di latar hutan 2 "hide" kategori *Looks* menyembunyikan sprite dari layar digunakan untuk sprite yang tidak diperlukan saat latar berada di hutan 2, seperti elemen dari level sebelumnya atau menu.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah memicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when backdrop switches to hutan 3" kategori *Events* kode ini akan dijalankan ketika latar belakang (backdrop) berubah menjadi "hutan 3." Artinya, aksi berikutnya hanya berlaku ketika backdrop aktif adalah "hutan 3", "go to x: 0 y: -150" kategori *Motion* ketika backdrop berubah menjadi "hutan 3," sprite akan langsung dipindahkan ke koordinat  $x = 0$  (tengah secara horizontal) dan  $y = -150$  (biasanya berada di bagian bawah layar), "forever" kategori *Control* blok ini membuat perintah di dalamnya

dijalankan terus-menerus tanpa berhenti, selama backdrop tetap "hutan 3", "set x to mouse x" kategori *Motion* dalam perulangan forever, posisi x sprite akan selalu diatur untuk mengikuti posisi x dari mouse. Artinya, sprite akan bergerak ke kiri atau kanan mengikuti pergerakan mouse.

## 12. Benar level 3



"when I receive level 3" *Events* kategori kode ini diaktifkan ketika sprite menerima pesan "level 3." Pesan ini biasanya dikirim dari bagian lain dalam program untuk menandakan bahwa permainan telah mencapai level 3. Ketika pesan ini diterima, kode berikutnya akan dijalankan, "show" kategori *Looks* setelah menerima pesan "level 3," sprite akan ditampilkan di layar. Jika sebelumnya sprite disembunyikan (menggunakan blok hide), maka perintah show akan membuat sprite terlihat lagi, "forever" kategori *Control* perintah forever membuat semua perintah di dalamnya dijalankan secara terus-menerus, selama kondisi sebelumnya (menerima pesan "level 3") tetap aktif, "play sound video permainan 1 until done" kategori *Sound* di dalam perulangan forever, suara "video permainan 1" akan diputar. "until done" memastikan suara tersebut diputar sepenuhnya sampai selesai sebelum melanjutkan ke perintah berikutnya.

"when I receive level 3" kategori *Events* kode ini akan dijalankan ketika sprite menerima pesan "level 3." Pesan ini biasanya dikirim oleh bagian lain dari program ketika permainan mencapai level 3. Begitu pesan diterima, kode berikutnya akan dieksekusi, "set nilai to 0" kategori *Variables* setelah menerima pesan "level 3," nilai dari *Variables* "nilai" akan diatur ke 0. Ini biasanya dilakukan untuk mereset skor atau variabel lainnya, memulai ulang permainan atau level, "forever" kategori *Control* blok ini membuat aksi di dalamnya dijalankan terus-menerus tanpa berhenti, selama kondisi sebelumnya (menerima pesan "level 3") terpenuhi, "wait pick random 1 to 2 seconds" kategori *Control* dan *Operators* dalam perulangan forever, program akan berhenti sejenak selama waktu acak antara 1 hingga 2 detik. Setiap kali perulangan berjalan, waktu jeda antara aksi berikutnya akan berbeda (acak), "create clone of

myself" kategori *Control* setelah menunggu selama waktu acak, clone (salinan) dari sprite ini akan dibuat. Jadi, setiap 1 hingga 2 detik, sebuah clone baru akan muncul di layar.

"when I receive level 3" *Events* kategori kode ini akan dijalankan ketika sprite menerima pesan dengan nama "level 3." Pesan ini biasanya dikirim dari bagian lain dalam program untuk menandakan bahwa permainan telah memasuki level 3. Begitu pesan ini diterima, kode berikutnya akan dijalankan, "set waktu to 30" kategori *Variables* setelah menerima pesan "level 3," *Variables* "waktu" akan diatur ke 30. Ini berarti waktu hitung mundur dimulai dari 30 detik, "forever" kategori *Control* bagian ini menunjukkan bahwa aksi berikutnya akan dijalankan terus-menerus selama kondisi sebelumnya (menerima pesan "level 3") terpenuhi, "wait 5 seconds" kategori *Control* di dalam perulangan forever, program akan berhenti selama 5 detik sebelum melanjutkan ke aksi berikutnya. Ini berarti perhitungan waktu dilakukan setiap 5 detik, "change waktu by -1" kategori *Variables* setelah menunggu selama 5 detik, *Variables* "waktu" akan dikurangi sebanyak 1. Ini berarti waktu hitung mundur berkurang 1 detik setiap kali, "if waktu < 1 then" kategori *Control*, *Variables*, dan *Operators* setelah mengurangi waktu, kode akan memeriksa apakah waktu sudah kurang dari 1 detik. Jika waktu menjadi kurang dari 1, maka blok kode berikutnya akan dijalankan, "switch backdrop to maaf coba lagi!" Kategori *Looks* jika waktu kurang dari 1, latar belakang (backdrop) akan berubah menjadi "maaf coba lagi!" Ini biasanya menunjukkan bahwa waktu telah habis dan pemain harus mencoba lagi, "stop all" kategori *Control* setelah latar belakang berubah, "stop all" akan menghentikan semua skrip yang sedang berjalan. Ini berarti permainan atau aktivitas yang sedang berlangsung akan berhenti, menandakan bahwa pemain gagal atau waktu telah habis.

"when I start as a clone" kategori *Control* kode ini akan dijalankan ketika sprite dibuat sebagai clone. Setiap clone yang dibuat akan menjalankan blok kode ini. Langkah 2: Menampilkan Clone dan Mengatur Posisi, "forever" kategori *Control* aksi berikutnya akan dijalankan terus-menerus selama clone masih ada, "show" kategori *Looks* Clone akan ditampilkan di layar. Jika sebelumnya clone disembunyikan (menggunakan hide), maka perintah show ini akan membuatnya terlihat, "switch costume to pick random 1 to 10" kategori *Looks* dan *Operators* Clone akan memilih tampilan (kostum) secara acak dari 1 hingga 10. Ini memberikan variasi pada tampilan setiap clone yang dibuat, "go to random position" kategori *Clone* akan diposisikan secara acak di layar, "set y to 500" kategori *Motion* Clone akan diposisikan pada y = 500 (biasanya di luar bagian atas layar), memulai pergerakan dari atas layar. Langkah 3: Pergerakan Clone, "repeat until y position < -150" kategori *Control*, *Motion* dan *Operators* perulangan ini akan terus berjalan sampai posisi y clone kurang dari -150 (di luar bagian bawah layar), "change y by -3" kategori *Motion* setiap kali perulangan berlangsung, posisi y akan dikurangi sebanyak 3, sehingga clone bergerak turun ke bawah layar. Langkah 4: Interaksi dengan "Kupu Kupu lv 3", "if touching kupu kupu lv 3?" Kategori *Control* dan *Sensing* jika clone menyentuh sprite "kupu kupu level 3,"



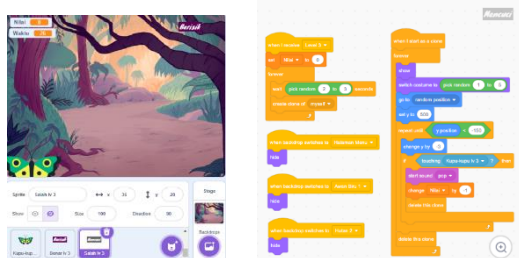
maka aksi berikutnya akan dijalankan, "start sound video permainan 1". Suara "video permainan 1" akan diputar untuk memberikan efek audio ketika clone menyentuh "kupu kupu lv 3", "change poin by 1" kategori *Variables* "poin" akan bertambah sebanyak 1 setiap kali clone menyentuh "kupu kupu lv 3", "delete this clone" kategori *Control* setelah interaksi dengan "kupu kupu level 3," clone akan dihapus. Langkah 5: Cek Kondisi Kemenangan, "if nilai > 10 then" kategori *Control, Operators* dan *Variables* jika nilai variabel "nilai" lebih besar dari 10, maka aksi berikutnya akan dijalankan, "switch backdrop to tepat sekali! Lanjutkan!" kategori *Looks* latar belakang (backdrop) akan berubah menjadi "tepat sekali! Lanjutkan!" untuk menunjukkan bahwa pemain telah mencapai target dan berhasil, "stop all" kategori *Control* semua skrip dalam proyek akan dihentikan. Ini biasanya menunjukkan bahwa level atau permainan selesai, "delete this clone" kategori *Control* setelah mencapai kondisi kemenangan, clone ini akan dihapus.

"when backdrop switches to [hutan 2]" kategori *Events* berfungsi memicu skrip tertentu saat latar belakang berubah menjadi hutan 2 digunakan untuk mengatur elemen atau aksi yang hanya terjadi di latar hutan 2, "hide" kategori *Looks* berfungsi menyembunyikan sprite dari layar. digunakan untuk sprite yang tidak diperlukan saat latar berada di hutan 2, seperti elemen dari level sebelumnya atau menu.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when backdrop switches to hutan 3" kategori *Events* blok ini berarti perintah di bawahnya akan dijalankan ketika latar belakang (backdrop) proyek berubah menjadi "hutan 3." Perintah ini hanya akan aktif jika latar belakang yang ditampilkan adalah "hutan 3", "hide" kategori *Looks* blok ini menyembunyikan sprite yang menjalankan perintah ini. Jadi, ketika latar belakang berubah menjadi "hutan 3," sprite tersebut akan disembunyikan dari layar.

### 13. Salah level 3



"when I receive level 3" kategori *Events* ini berarti ketika objek menerima pesan "level 3", kode berikutnya akan dijalankan, "set nilai to 0" kategori *Variables* yang disebut "nilai" akan diatur ke 0, "Forever" kategori *Control* menandakan bahwa perintah di dalam blok ini akan dijalankan terus-menerus selama program berjalan, "wait pick random 2 to 3 seconds" kategori *Control* dan *Operators* program akan berhenti sementara selama waktu acak antara 2 hingga 3 detik, "create clone of myself" kategori *Control* setelah waktu tunggu selesai, program akan membuat salinan (klon) dari objek tersebut, sehingga ada dua objek yang sama.

"when backdrop switches to halaman menu" kategori *Events* kode ini adalah pemicu yang akan aktif ketika latar belakang (backdrop) berubah menjadi "halaman menu." Jadi, aksi berikutnya akan dijalankan hanya jika latar belakang saat itu adalah "halaman menu", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (tidak terlihat di layar). Fungsi ini biasanya digunakan untuk menghilangkan elemen atau objek dari tampilan ketika tidak diperlukan lagi, seperti ketika pemain masuk ke halaman menu.

"when backdrop switches to awan biru 1" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "awan biru 1." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "awan biru 1", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

"when backdrop switches to hutan 2" kategori *Events* bagian ini adalah pemicu yang mengaktifkan kode ketika latar belakang (backdrop) berubah menjadi "hutan 2." Kode berikutnya hanya akan dijalankan saat latar belakang saat itu adalah "hutan 2", "hide" kategori *Looks* setelah latar belakang berubah, sprite yang menjalankan kode ini akan disembunyikan (hilang dari tampilan layar). Fungsi ini berguna untuk menghilangkan elemen atau objek tertentu dari tampilan ketika tidak diperlukan lagi.

"when I start as a clone" kategori *Control* kode ini dijalankan setiap kali klon dari objek ini dibuat, "forever" kategori *Control* menandakan bahwa perintah di dalam blok ini akan dijalankan terus-menerus selama klon tersebut masih ada, "show" kategori *Looks* Clone objek akan muncul di layar, "switch costume to pick random 1 to 5" kategori *Kostum* klon akan dipilih secara acak dari pilihan 1 hingga 5. Kostum ini digunakan untuk memberikan variasi tampilan pada klon, "go to random position" kategori *Motion* Klon akan dipindahkan ke posisi acak di layar, "set y to 500" kategori *Motion* Klon akan diposisikan pada koordinat y = 500 (posisi vertikal di atas layar), "repeat until y position < -150" kategori *Control*, *Motion*, dan *Operators* Klon akan terus bergerak sampai posisi vertikal (y) mencapai nilai kurang dari -150. Proses ini dilakukan dalam perulangan, "change y by -3" kategori *Sensing* setiap kali perulangan dijalankan, posisi y klon akan dikurangi 3, sehingga klon akan bergerak ke bawah, "If touching "kupu kupu lv 3?" kategori *Control* dan *Sensing* jika

klon menyentuh objek yang bernama "kupu kupu lv 3", kode berikutnya akan dijalankan, "Start sound pop: Suara "pop" kategori *Sound* akan diputar ketika klon menyentuh objek tersebut, "Change nilai by -1" kategori *Variables* "nilai" akan berkurang 1 setiap kali klon menyentuh objek "kupu kupu lv 3", "delete this clone kategori *Control* setelah peristiwa ini terjadi, klon akan dihapus dari layar.

## SIMPULAN

Desain game kata sifat berbasis scratch untuk melatih kemampuan mengidentifikasi kata sifat siswa sekolah dasar memiliki blok kode 1) Sprite Mangkok, 2) Tombol 2, 3) Tombol 3, 4) Tombol 4, 5) Dinosaurus karakter utama pada level 1, 6) Benar level 2, 7) Salah level 1, 8) Monyet karakter utama pada level 2, 9) Benar level 2, 10) Salah level 2, 11) Kupu-kupu karakter utama level 3, 12) Benar level 3, 13) Salah level 3.

## REFERENSI

- Abdussamad, Z. (2022). *Buku Metode Penelitian Kualitatif*.
- Afriani, T. (2022). *Membuat Game Scratch Pertamaku*. Bhuana Ilmu Populer.
- Arianti, F. F., Sutrimah, S., & Hasanudin, C. (2020). Flipped classroom dan aplikasi schoology: Analisis keterampilan menulis teks biografi. *Tabasa: Jurnal Bahasa, Sastra Indonesia, dan Pengajarannya*, 1(2), 165-186. <https://doi.org/10.22515/tabasa.v1i2.2591>.
- Asiah, N. (2018). Pembelajaran calistung Pendidikan anak usia dini dan ujian masuk calistung sekolah dasar di Bandar Lampung. *Terampil: Jurnal Pendidikan Dan Pembelajaran Dasar*, 5(1), 19-42. <https://ejournal.radenintan.ac.id/index.php/terampil/article/view/2746>.
- Azzam, A. A. M. (2022). *Fiqh Muamalat: Sistem Transaksi dalam Fiqh Islam*. Amzah.
- Damayanti, R. (2015). Penggunaan deksis semantik dalam cerpen Siluet Jingga Karya Anggi P. *Jurnal Ilmiah Buana Bastra: Bahasa, Susastra, dan Pengajarannya*, 2(2), 175-187. <https://jurnal.unipasby.ac.id/index.php/bastra/article/view/232>.
- Fadli, M. R. (2021). Memahami desain metode penelitian kualitatif. *Humanika, Kajian Ilmiah Mata Kuliah Umum*, 21(1), 33-54. [https://scholar.archive.org/work/a3mjph7zgrchfhdz5bah2o66de/access/w/wayback/https://journal.uny.ac.id/index.php/humanika/article/download/38075/pdf\\_1](https://scholar.archive.org/work/a3mjph7zgrchfhdz5bah2o66de/access/w/wayback/https://journal.uny.ac.id/index.php/humanika/article/download/38075/pdf_1).
- Fendrik, M., Putri, D. F., Pebriana, P. H., Sidik, G. S., & Ramadhani, D. (2022). The analisis kecenderungan gaya belajar siswa sekolah dasar. *Jurnal Pendidikan dan Konseling (JPDK)*, 4(3), 793-809. <https://doi.org/10.31004/jpdk.v4i3.4094>.
- Gultom, U., & Murpratomo, J. (2018). Sistem pelayanan jemaat berbasis web. *JISAMAR (Journal of Information System, Applied, Management, Accounting and*

- Research*), 2(1), 55-62.  
<http://journal.stmikjayakarta.ac.id/index.php/jisamar/article/view/28>.
- Hakim, A. R. (2023). Konsep landasan dasar pendidikan karakter di Indonesia. *Journal on Education*, 6(1), 2361-2373. <https://doi.org/10.31004/joe.v6i1.3258>.
- Hamzah, R. E. (2015). Penggunaan media sosial di kampus dalam mendukung pembelajaran pendidikan. *WACANA: Jurnal Ilmiah Ilmu Komunikasi*, 14(1), 45-70. <http://journal.moestopo.ac.id/index.php/wacana/article/view/89>.
- Hansopaheluwakan, S., Ifadah, E., Baka, C., Putri, L. A., Karimuddin, K., Suningsih, S. S. S., & Judijanto, L. (2024). *Basic English Grammar: Panduan Dasar Mahir Berbahasa Inggris*. PT. Green Pustaka Indonesia.
- Harimurti, Y., & Udariansyah, D. (2023). Implementasi Service EC2 & S3 Amazon Web Service Pada Niche Blog Menggunakan Metode SDLC. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(2), 675-685. <https://doi.org/10.30865/klik.v4i2.1192>.
- Holis, A. (2017). Belajar melalui bermain untuk pengembangan kreativitas dan kognitif anak usia dini. *Jurnal Pendidikan UNIGA*, 10(1), 23-37. <http://dx.doi.org/10.52434/jp.v10i1.84>.
- Isnaini, M., Fujiaturahman, S., Utami, L. S., Zulkarnain, Z., Anwar, K., Islahudin, I., & Sabaryati, J. (2021). Pemanfaatan aplikasi scratch sebagai alternatif media belajar siswa "z generation" untuk guru-guru sdn 1 labuapi. *SELAPARANG: Jurnal Pengabdian Masyarakat Berkemajuan*, 5(1), 871-875. <https://doi.org/10.31764/jpmb.v5i1.6554>.
- Jalaluddin, J. (2024). Perbandingan Fikih tentang Gharar: Analisis Praktik Gharar dalam Transaksi Ekonomi Islam. *Jurnal Maps (Manajemen Perbankan Syariah)*, 8(1), 50-62. <http://jurnal.masoemiversity.ac.id/index.php/maps/article/view/880>.
- Julianti, S. (2021). *A Practical Guide to Corrugated Box: Material, Teknologi, dan Aplikasi Kardus*. Gramedia Pustaka Utama.
- Kartika, D., & Budaya, J. S. J. F. I. (2017). *Analisi konstruktif kata sifat (Keiyoshi) Bahasa Jepang dan Bahasa Indonesia ditinjau secara Gramatikal serta Pengajarannya*. Universitas Bung Hatta. Padang.
- Kurniawan, A. D. D. (2013). Pengembangan buku siswa untuk meningkatkan proses dan hasil belajar kompetensi dasar cornflake cookies pada siswa Tunagrahita SMA-LB Negeri Gedangan, Sidoarjo. *Jurnal Tata Boga*, 2(1). <https://ejournal.unesa.ac.id/index.php/jurnal-tata-boga/article/view/1072/787>.
- Liliweri, A. (2017). *Komunikasi antar personal*. Prenada Media.

- Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, A. (2020). *Computational thinking pemecahan masalah di abad ke-21*. Madiun: Perpustakaan Nasional: Katalog Dalam Terbitan (KDT).
- Mashar, R. (2015). *Emosi anak usia dini dan strategi pengembangannya*. Kencana.
- Mufidah, T. H., & Majid, N. W. A. (2024). Pengaruh peningkatan computational thinking Siswa Kelas 5 Melalui Pembelajaran Dasar Coding. *Buletin Literasi Budaya Sekolah*, 22-37. <https://doi.org/10.23917/blbs.v6i1.4231>.
- Muzzamil, F. (2021). Pengaruh lingkungan terhadap perkembangan sosial emosional anak. *MURANGKALIH: Jurnal Pendidikan Anak Usia Dini*, 2(02). <https://doi.org/10.35706/murangkalih.v2i02.5811>.
- Mustafida, F. (2013). Kajian Media Pembelajaran Berdasarkan Kecenderungan Gaya Belajar Peserta Didik SD/MI. *Madrasah: Jurnal Pendidikan dan Pembelajaran Dasar*, 6(1), 20. <https://doi.org/10.18860/jt.v6i1.3291>.
- Nabilah, A. P., Alindra, A. L., Nurhikmah, I., Fauziyah, N. N., Herlina, P., Febriyanti, R., & Prayoga, R. (2024). Penggunaan media scratch meningkatkan motivasi belajar dan kemampuan berpikir kreatif matematis siswa. *Jurnal Pendidikan Tambusai*, 8(1), 1975-1986. <https://doi.org/10.31004/jptam.v8i1.12694>.
- Nuzuli, A. K. (2022). *Dasar-Dasar Penulisan Karya Ilmiah*. Jejak Pustaka.
- Olivia, C. D., & Nirmalasari, Y. (2022). Analisis penggunaan kata bantu bilangan dalam kalimat bahasa mandarin pada pabrik tambahan 副刊 (Fù Kān) Koran Harian Nusantara 千島日報 (Qiāndǎo Ribào) Edisi Bulan Juni 2021. *KLAUSA (Kajian Linguistik, Pembelajaran Bahasa, dan Sastra)*, 6(2), 39-56. <https://doi.org/10.33479/klausu.v6i2.602>.
- Peniarsih, P. (2021). *Cloud computing technologies and business opportunities*. *JURNAL TEKNOLOGI INDUSTRI*, 1.
- Pertiwi, K. D., & Kurniawan, Y. (2016). Spesifikasi Kebutuhan Perangkat Lunak Sistem Informasi Akademik Universitas Ma Chung Malang. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 7(2), 627-634. <http://jurnal.umk.ac.id/index.php/simet/article/view/775>.
- Perkasa, R. A. E., & Wantoro, J. (2024). Game edukasi interaktif sejarah kerajaan hindu-buddha menggunakan Platform Scratch. *Edumatic: Jurnal Pendidikan Informatika*, 8(1), 36-45. <https://doi.org/10.29408/edumatic.v8i1.25161>.
- Priharsanto, A. S. (2024). Pengembangan aplikasi kesehatan mental berbasis kotlin: Studi kasus aplikasi Hugme. *Jurnal Kajian Teknik Elektro*, 9(2), 102-108. <https://journal.uta45jakarta.ac.id/index.php/JKTE/article/view/7690>.

- Rahayu, S., & Rahayu, P. A. (2017). Perancangan sistem informasi pengelolaan keuangan berbasis web di sekolah menengah kejuruan Islam Atturmudziyyah Garut. *Jurnal Algoritma*, 14(2), 538-545. <https://jurnal.itg.ac.id/index.php/algoritma/article/view/565>.
- Rahman, T. (2017). Teks dalam kajian struktur dan kebahasaan. CV. Pilar Nusantara.
- Rismayanthi, C. (2013). Mengembangkan keterampilan gerak dasar sebagai stimulasi motorik bagi anak taman kanak-kanak melalui aktivitas jasmani. *Jurnal Pendidikan Jasmani Indonesia*, 9(1). <https://doi.org/10.21831/jpji.v9i1.3065>.
- Rizaldi, A., Pranatawijaya, V. H., & Putra, P. B. A. A. (2021). Penerapan antrian dan pemesanan online di aplikasi pearl salon and barnershop berbasis mobile. *Journal of Information Technology and Computer Science*, 1(1), 1-9. <https://ejournal.upr.ac.id/index.php/jcoms/article/view/2384>.
- Rukompoy, T. R. (2018). Kata sifat bahasa Inggris dalam jurnal English Teaching Forum. *Jurnal elektronik fakultas sastra Universitas SAM Ratulangi*, 2(1). <https://ejournal.unsrat.ac.id/index.php/jefs/article/view/19364>.
- Rosyidah, U., Hasanudin, C., & Amin, A. K. A. (2021). Kajian frasa pada novel trauma karya boy candra. *Jurnal Ilmiah SEMANTIKA*, 3(01), 10-20. <https://doi.org/10.46772/semantika.v3i01.460>.
- Sadli, S. (2010). *Berbeda tetapi setara: pemikiran tentang kajian perempuan*. Penerbit Buku Kompas.
- Sandy, M., & Lazuardi, S. N. (2024). Rancang bangun aplikasi data kepegawaian berbasis web menggunakan metode waterfall. *Jurnal Tera*, 4(1), 1-10. <http://jurnal.undira.ac.id/jurnaltera/article/view/287>.
- Setiawan, Z., Pustikayasa, I. M., Jayanegara, I. N., Setiawan, I. N. A. F., Putra, I. N. A. S., Yasa, I. W. A. P., ... & Gunawan, I. G. D. (2023). *Pendidikan multimedia: Konsep dan Aplikasi pada era revolusi industri 4.0 menuju society 5.0*. PT. Sonpedia Publishing Indonesia.
- Sopiyan, A., & Darajatun, R. A. (2024). Analisis Situasional Dan Perancangan Sistem Informasi Keuangan Pada Pt Zmi. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(1). <http://dx.doi.org/10.23960/jitet.v12i1.3753>.
- Sulaksono, D. (2019). Analisis Aspek Gramatikal dan Leksikal Iklan Kecil dalam Surat Kabar Solopos. *JISABDA: Jurnal Ilmiah Sastra dan Bahasa Daerah, serta Pengajarannya*, 1(2), 13-21. <http://journal.upgris.ac.id/index.php/jisabda/article/view/4740>.
- Supit, D., Melianti, M., Lasut, E. M. M., & Tumbel, N. J. (2023). Gaya belajar visual, auditori, kinestetik terhadap hasil belajar siswa. *Journal on Education*, 5(3), 6994-7003. <https://doi.org/10.31004/joe.v5i3.1487>.

- Suyanto, B. (2015). *Metode penelitian sosial: Berbagai alternatif pendekatan*. Prenada Media.
- Utami, L. A., & Yuningsih, P. D. (2024). Sistem Informasi Online Booking Berbasis Web Pada Pheo Studi Salon. *Jurnal Teknoinfo*, 18(1), 193-200. <https://ejournal.teknokrat.ac.id/index.php/teknoinfo/article/view/2665>.
- Wahyuni, S., & Harun, M. (2018). Analisis struktur fisik dan struktur batin puisi anak dalam majalah potret anak cerdas. *Master Bahasa*, 6(2), 115-125. <https://jurnal.usk.ac.id/MB/article/view/11601>.
- Widarmanto, T. (2017). *Pengantar Jurnalistik Panduan Awal Penulis dan Jurnalis Edisi Revisi*. Araska Publisher.
- Widiyanto, W. W. (2018). Analisa metodologi pengembangan sistem dengan perbandingan model perangkat lunak sistem informasi kepegawaian menggunakan waterfall development model, model prototype, dan model rapid application development (rad). *Jurnal Informa: Jurnal Penelitian Dan Pengabdian Masyarakat*, 4(1), 34-40. <https://informa.poltekindonusa.ac.id/index.php/informa/article/view/34>.
- Winarno, M. E. (2013). *Metodologi penelitian dalam pendidikan jasmani*. Um press.
- Zandroto, W. A. S., Lubis, N. A. P., Pangaribuan, W., & Mudjisusaty, Y. (2024). Penggunaan frasa kata benda dalam kalimat ilmiah. *Dharmas Education Journal (DE\_Journal)*, 4(3), 195-202. [https://ejournal.undhari.ac.id/index.php/de\\_journal/article/view/1271](https://ejournal.undhari.ac.id/index.php/de_journal/article/view/1271).