

BAB II

DASAR TEORI DAN TINJUAN PUSTAKA

2.1. Tinjauan Terhadap Penelitian Yang Sudah Dilakukan

Berikut ini hasil kajian penelitian yang pernah dilakukan dengan berbagai pendekatan dalam pengembangan aplikasi pembelajaran berbasis android:

Pengembangan Permainan Edukasi Katelu (Klasifikasi komponen computer) Berbasis Android dengan *Tools Unity 3D Game Engine* (Rahman, Afif Ftchur, 2017). Penelitian ini bertujuan : mengembangkan *game* edukasi katelu berbasis android, memperoleh hasil *game* edukasi yang sesuai standar ISO 9126 dan *game* yang layak sebagai *game* pendidikan. Sistem pendekatan yang digunakan adalah *research and Develompemnt (R&D)* dan model pengembangan multimedia dari Lee dan Owens. Hasil Aplikasi : *game* berjalan pada platform android dan bergenre *quiz and puzzle* dengan memiliki tingkatan 3 level yang berbeda serta memiliki fitur nilai tertinggi dan rekap nilai.

Fransiska, T (2019), penelitian ini menghasilkan aplikasi pembelajaran pengenalan nama-nama hewan darat dan air untuk usia 6 tahun, agar interaktif dan dapat meningkatkan minat serta motivasi anak-anak dalam belajar dan menghibur juga mendidik anak-anak agar bisa menjadikan anak-anak lebih mudah belajar dengan cara menggunakan aplikasi di android atau *mobile*. Aplikasi ini dikembangkan menggunakan bahasa pemrograman *java* dan *database* menggunakan *Sqlite* dan menggunakan *Android Studio*.

Yandaka Ganesha Putra (2017), Pembuatan *Game* Balap Kucing dengan *Unity* berbasis Android Pembuatan *game* balap lari kucing dengan aplikasi *Unity* dan blender 3D, pembuatan *game* dengan *script-script* dan fitur *save* dan *load* untuk membaca dan menyimpan data pemain, *game* dengan banyak *scene* tanpa kehilangan data ketika *game* dijalankan dan *game* harus berpindah *scene* ke *scene* lain. Hal ini dapat mengurangi resiko munculnya bug.

Andika Rifqi Istanto (2022) Sistem Perhitungan Omset Dan Penggajian Pada Cantik Salon Kotagede Menggunakan Laravel. Tujuan dari penelitian ini dapat menghitung omset dan pengeluaran per hari dan membantu proses penggajian karyawan serta membuat laporan bulanan.

Nur Fitria Amalia (2021) Pengembangan Game Edukasi "Math Adventure" Berbasis Android untuk Anak Usia Dini mencakup pembelajaran berbasis game menggunakan elemen permainan tujuan pendidikan, meningkatkan motivasi dan keterlibatan siswa. Pengembangan game edukasi mengikuti proses perencanaan hingga evaluasi, menggunakan alat seperti Unity 3D dan C#. Teori *konstruktivisme* dan Piaget menekankan pentingnya interaksi dan permainan simbolis dalam belajar anak usia dini. Matematika adalah keterampilan dasar penting, dan game dapat membuat belajar matematika menyenangkan. Standar ISO 9126 memastikan kualitas perangkat lunak dalam aspek fungsionalitas, keandalan, kegunaan, efisiensi, pemeliharaan, dan *portabilitas*.

Satria Dwijo Pranowo (2024) Pembuatan game dengan judul "*Bubble Jump Berbasis Android*", tujuan diharapkan memberikan kontribusi dalam pembuatan game 2D, permainan pemain melewati rintangan gedung dan awan dan koin ajak

dengan level kesulitan yang ditentukan dengan menghasilkan game berbasis android, menggunakan pendekatan *Unity 2D* dan *C#* melalui system UML.

Beberapa penelitian di atas dibuat tabel perbandingan terkait dengan metode dan hasil yang diperoleh sebuah simpulan kebaharuan dari penelitian yang dilakukan. Adapun Tabel Tinjauan pustaka dapat dilihat pada tabel 2.1.

Tabel 2. 1. Tinjauan Pustaka

No	Penulis	Judul	Tujuan Penelitian	Hasil Aplikasi
1	Rahman, Afif Ftchur (2017)	Pengembangan Permainan Edukasi Katelu (Klasifikasi Komponen Komputer) Berbasis Android dengan Unity 3D	Mengembangkan game edukasi Katelu berbasis Android, sesuai standar ISO 9126, dan sebagai game pendidikan menggunakan pendekatan R&D dan model Lee dan Owens.	Game berjalan pada platform Android, bergenre quiz and puzzle, memiliki 3 level berbeda, fitur nilai tertinggi, dan rekap nilai.
2	Fransiska, T 2019	Aplikasi Pembelajaran Pengenalan Nama-nama Hewan Darat dan Air untuk Usia 6 Tahun	Aplikasi Pembelajaran Pengenalan Nama-nama Hewan Darat dan Air untuk Usia 6 Tahun	Aplikasi Pembelajaran Pengenalan Nama-nama Hewan Darat dan Air untuk Usia 6 Tahun
3	Yandaka Ganesha Putra 2017	Pembuatan Game Balap Kucing dengan Unity berbasis Android	Mengembangkan game balap lari kucing menggunakan Unity dan Blender 3D, dengan fitur save dan load data, memastikan data tidak hilang saat berpindah scene.	Game memiliki banyak scene tanpa kehilangan data, mengurangi risiko munculnya bug.
4	Andika Rifqi Istanto 2022	Sistem Perhitungan Omset dan Penggajian pada Cantik Salon Kotagede Menggunakan	Menghitung omset dan pengeluaran per hari, membantu proses penggajian karyawan, serta membuat laporan bulanan.	Sistem menggunakan framework Laravel, memudahkan perhitungan

No	Penulis	Judul	Tujuan Penelitian	Hasil Aplikasi
		Laravel		omset, penggajian karyawan, dan pembuatan laporan bulanan.
5	Nur Fitria Amalia 2021	Pengembangan Game Edukasi "Math Adventure" Berbasis Android untuk Anak Usia Dini	Mengembangkan game edukasi berbasis Android yang interaktif untuk membantu anak usia dini belajar matematika, meningkatkan motivasi belajar, menggunakan pendekatan Unity 3D dan C#.	Game berbasis Android dengan fitur interaktif, animasi menarik, dan level-level menantang, sesuai standar ISO 9126.
6	Satria Dwijo Pranowo 2023	Pembuatan Game "Bubble Jump" Berbasis Android	Mengembangkan game 2D berbasis Android dimana pemain melewati rintangan gedung, awan, dan koin dengan level kesulitan yang bervariasi, menggunakan pendekatan Unity 2D dan C# melalui sistem UML.	Game berbasis Android dengan pemain melewati rintangan dan koin, menggunakan Unity 2D dan C# melalui sistem UML.

Berdasarkan dari beberapa penelitian Fransiska (2019) dengan pendekatan *java*, *database*, *sqlite* dan android studio dengan menghasilkan aplikasi iteraktif pembelajaran pengenalan nama-nama hewan darat dan laut, kemudian penelitian Putra (2017) dengan pendekatan *game* menggunakan aplikasi *Unity* dan Blender 3D menghasilkan *game* dengan *Script* dan fitur save dan load untuk membaca dan menyimpan data pemain. Penelitian lainnya adalah Istanto (2022) yang menggunakan bahasa pemrograman *php* dan *database mysql* dengan relasi antar tabel, DAD, dan DAD level 1. Pengembangan atas beberapa penelitian yang

sudah ada tersebut maka penelitian ini akan mengembangkan sebuah permainan/*game* dengan topik “*Bubble Jump*” dengan model permainan pemain akan melewati rintangan antara gedung dan awan berdasarkan dari level yang ditentukan, atas dasar ini maka akan dihasilkan sebuah aplikasi *game bubble Jump* berbasis android, sedangkan dalam pendekatan untuk mengembangkan *game* tersebut menggunakan *Unity 2D* dan *C#* melalui perancangan sistem UML.

2.2. Dasar Teori

2.2.1. *Game engine*

Game Engine adalah perangkat lunak yang menyediakan seperangkat alat dan layanan membantu pengembangan dan pembuatan video game. Game engine biasanya mencakup berbagai fitur seperti rendering grafis (2D atau 3D), fisika simulasi, deteksi benturan, scripting, kecerdasan buatan, manajemen memori, dan audio. Dengan menggunakan game engine, pengembangan dapat menghemat waktu dan sumber daya karena banyak aspek teknis dari pembuatan game sudah ditangani oleh perangkat lunak ini. (Unreal Engine Official Website,2020).

2.2.2. *Unity*

Unity (Blackman, dalam Khaerudin et al, 2021) adalah *software* yang terpilih. *Unity* merupakan aplikasi yang ideal sebagai awal dari pemodelan *game* yang di dalamnya tersedia fitur-fitur yang tinggi dengan kemampuan *authoring multiplatform*. *Unity* menjadi sebuah *software* yang baik dengan memiliki banyak perpaduan dengan aplikasi profesional, editor dengan segala kemudahan dan *user interface* yang sederhana. Berikut ini menu *interface unity*:

a. *Asset*

Asset adalah tempat penyimpanan dalam *unity*, suara, gambar, video, tekstur, semua yang ingin dipakai dalam *unity* disimpan dalam *asset*.

b. *Scenes*

Scenes adalah sebuah area yang berisikan konten-konten dalam *game*, seperti membuat *level*, membuat menu, tampilan tunggu, dsb.

c. *Game object*

Ketika semua barang di dalam *assets* dipindahkan ke dalam *scenes*, maka benda tersebut berubah menjadi *game object*, di mana benda tersebut bisa di gerakan, diatur ukuran dan diatur rotasinya.

d. *Components*

Components dapat di masukan ke dalam *game object* untuk menciptakan reaksi baru, seperti *collsion*, memunculkan partikel, dsb.

Intinya *components* membuat reaksi baru di dalam *game object*.

e. *Script*

Script yang dapat digunakan dalam *unity* ada 3, yaitu *javascript*, *C#* dan *BOO*. Tetapi *unity* tidak menyediakan cara untuk menggunakan *script* tersebut. *Script* dalam *unity* tidaklah begitu sulit karena *script* tidaklah bertele-tele, melainkan langsung menunjuk kepada aksi yang ingin diberikan. Penggunaan *Script* dalam *unity* harus menggunakan program lain yang disediakan oleh *Unity*.

f. Prefabs

Prefabs adalah tempat untuk menyimpan satu jenis *game objects*, sehingga mudah untuk diperbanyak. *Prefabs* juga mempermudah dalam pembuatan objek-objek yang kompleks, tetapi tujuan utama *prefabs* adalah untuk mempermudah memunculkan banyak objek dalam seketika.

2.2.3. CoreDRAW

CoreDRAW merupakan salah satu pengolah grafis dengan basis *vector*, Di mana unsur dasar yang mendasarinya adalah garis. Keuntungan dari *vector* ini adalah gambar akan mempunyai ukuran kapasitas *file* yang *relative* kecil apabila dibandingkan dengan pengolah grafis berbasis *bitmap*. Juga tidak ada penurunan kualitas jika gambar diperbesar (Lpkpalbarcollege.com, 2020).

CoreDRAW adalah aplikasi yang berfungsi sebagai editor grafik *vector* yang menghasilkan gambar visual. Aplikasi ini fokus pada editor gambar, sehingga banyak dipakai oleh pengguna dalam bidang *advertising*, desain visual, percetakan, dan bidang lainnya yang memerlukan format visualisasi (Budiarta, Sila, 2022). Beberapa kelebihan *coreDRAW* dibandingkan aplikasi lain dengan fungsi yang sama, antara lain (Budiarta, Sila, 2022):

1. Kemampuan mengolah garis dan warna yang cukup akurat
2. Dipenuhi berbagai jenis *font* untuk menunjang kreativitas dan imajinasi dalam mengolah brosur, *pamphlet*, sampul buku, dan sebagainya
3. Tingkat kejelasan dan spesifikasi warna yang mendetail terhadap desain yang diinginkan

4. Keakuratan dalam mendesain, misalnya dari sisi garis, lengkungan garis, sudut serta kerapatan garis.

2.2.4. Android

Android adalah perangkat lunak gratis dengan sumber terbuka, dalam artian Google memperbolehkan pengguna untuk mengembangkan sistem operasi tersebut. Android memiliki toko aplikasi bernama *Google Play Store* yang bebas *download* aplikasi atau *game* (Bintara, 2023).

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti smartphone dan tablet. Sedangkan, pendapat berbeda mengatakan android adalah suatu sistem operasi pada handphone yang bersifat terbuka dan berbasis Linux.(Hertiwi, Muhallim, dan Sulaeman 2022) jadi menurut penulis android merupakan sistem yang dirancang khusus untuk smartphone berbasis Linux.

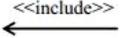
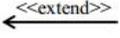
Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface (API)* yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga (Iqbal et al, 2015).

A. Use Case Diagram

Use case diagram adalah representasi visual dari interaksi antara aktor(pengguna atau sistem eksternal) dan sistem untuk mencapai tujuan tertentu.

Diagram ini digunakan untuk menangkap persyaratan fungsional dari suatu sistem dan menjelaskan interaksi antara aktor dan use case (fungsi sistem). (Guru99, "UML Use Case Diagram Tutorial with Example," 2020) Simbol-simbol yang ada pada diagram *use case* dapat di lihat pada table 2.2.

Tabel 2. 2. Simbol *diagram use case*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

(<https://www.dicoding.com/blog/contoh-use-case-diagram/>)

B. Activity Diagram

Menurut Munawar (Aqli, A., 2021) *Activity Diagram* adalah untuk menangkap tingkah laku dinamis dari system dengan cara menunjukkan aliran pesan dari suatu aktivitas ke aktivitas lainnya.

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini adalah bahwa diagram

aktivitas menggambarkan aktivitas sistem apa yang bukan dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang ada pada diagram aktivitas dapat di lihat pada table 2.3.

Tabel 2. 3. Simbol activity diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(<https://www.dicoding.com/blog/apa-itu-activity-diagram/>)

C. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan sequence diagram maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstantiasi menjadi itu. Membuat Sequence diagram

juga dibutuhkan untuk melihat skenario yang ada pas use case. Simbol Sequence diagram data dilihat pada Tabel 2.4.

Tabel 2. 4. Simbol sequence diagram

	User atau orang, proses atau sistem yang berinteraksi dengan sistem informasi untuk menciptakan
	objek berinteraksi pesan ditunjukkan
	objek yang ditunjukkan aktif dan interaksi, semua yang berhubungan dengan saat ini, ini adalah langkah yang tidak.
	Mendeklarasikan sebuah objek adalah subjek, kepala ke arah Panah untuk objek yang dibuat
	Menyatakan sebuah objek memanggil operasi / metode
	Menyatakan objek untuk mengirim masukan data / informasi ke objek lain.
	Menyatakan sebuah objek yang menjalankan operasi atau menghasilkan kembali ke objek tertentu.

2.2.5. Android Software Development Kit (SDK)

Android SDK adalah kumpulan software yang berisi mengenai pustaka, *debugger* (alat pencarian program), *emulator* (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan. Android SDK dapat diunduh secara gratis di <http://developer.android.com/sdk/>. Menurut Meier (2012), Android SDK

sudah termasuk *tools* yang dibutuhkan dalam mengembangkan aplikasi, melakukan pengujian, dan melakukan debug seperti:

a. Android API

Android SDK merupakan inti dari Android API *library* yang menyediakan para pengembang untuk mengakses kumpulan *library* pada Android. *Libraries* yang digunakan oleh Google merupakan *libraries* yang sama yang digunakan untuk membuat *native application*.

b. *Development Tools*

Pada SDK sudah termasuk beberapa *tools* yang digunakan untuk melakukan *compile* dan *debug* sehingga aplikasi dapat dijalankan pada perangkat Android.

c. *Android Virtual Device Manager dan Emulator*

Pada SDK sudah termasuk emulator untuk mensimulasikan aplikasi yang dikembangkan sebelum di-*install* pada perangkat.

d. *Full Documentation*

Android SDK menyediakan informasi dari setiap *package* dan cara menggunakannya dalam bentuk dokumentasi di dalam *source code*. Selain itu, terdapat juga *Android reference documentation* dan *developer guide* yang menjelaskan cara memulai dalam mengembangkan aplikasi Android, menyertakan *best practices*, dan menyediakan pembahasan *framework* secara mendalam.

e. *Sample Code*

Android SDK menyediakan sampel program yang menunjukkan beberapa fungsi yang bisa dilakukan pada Android dan cara menggunakan fitur pada API.

f. *Online Support*

Android memiliki komunitas untuk para pengembang aplikasi. *Google groups* merupakan salah satu forum dari kumpulan para pengembang aplikasi Android. Selain itu, *Stack Overflow* merupakan tempat di mana para pengembang aplikasi mendapatkan jawaban khususnya bagi *programmer* pemula.

2.2.6. *Star UML*



Gambar 2. 1. Aplikasi *Star UML*

StarUML merupakan sebuah *software* pemodelan yang mendukung UML (*Unified Modelling Language*). StarUML bisa memaksimalkan produktifitas dan kualitas dari sebuah *software* project karena mendukung notasi UML 2.0 dan juga mendukung pendekatan MDA (*Model Driven Architecture*). <https://staruml.io/>

StarUML membuat perbedaan konseptual yang lebih jelas antara *models*, *views* dan *diagrams*. *Model* adalah sebuah elemen yang memuat informasi untuk model *Software*. *View* merupakan elemen yang memuat informasi di dalam model. *Diagram* adalah suatu koleksi dari elemen yang memberikan pemikiran *user* di dalam mendesain secara spesifik.

A. *Project*

Project merupakan unit manajemen dasar di dalam StarUML. suatu *Project* dapat mengatur satu ataupun lebih model *software*. *Project* merupakan *top-level*

package di dalam beberapa model *software*. Secara umum, suatu *project* di dalam satu *file*. struktur *project* di dalam *Star UML* di mana sebuah *project* terdiri dari dan mengatur sub-elemen seperti pada Tabel 2.5

Tabel 2. 5. *Struktur project dalam star uml*

<i>Project sub-elemen</i>	Deskripsi
Model	Elemen yang mengatur suatu model software
<i>Subsystem</i>	Elemen yang mengatur model-model yang tercakup dalam suatu sub sistem
<i>Package</i>	Elemen yang paling umum untuk mengatur elemen

<https://kosemisme.blogspot.com//aplikasi-star-uml.html>

File project disimpan ke dalam format XML dengan *extension* ".uml". semua model, *view* dan *diagrams* yang di buat dengan *StarUML* disimpan dalam satu *file* project berisikan informasi sebagai berikut ini :

1. UML *profile* yang digunakan dalam *project*.
2. *File* unit yang *diferensi* oleh *project*
3. Informasi untuk semua model yang ada di dalam *project*
4. Informasi untuk semua *diagrams* dan *views* yang ada di dalam *project*.

B. Units

Terdapat beberapa kasus di mana satu *project* perlu disimpan di dalam beberapa *file-file* kecil sehingga para pengembang dapat bekerja di dalam satu project secara bersamaan. di dalam kasus ini suatu *project* dapat mengatur bermacam-macam unit. suatu unit mempunyai struktur hirarki dan berisikan beberapa sub-unit. Unit disimpan sebagai ".uml" *file* dan beberapa mengacu pada *file project* (.uml) atau unit *file* lainnya (.unt).

Komposisi unit terdiri dari *package*, *subsystem* dan elemen model yang dapat membentuk satu unit. semua elemen di bawah jenis elemen *package* ini disimpan di dalam masing-masing *file unit* (.unt)

C. Module

Modul adalah suatu *package* yang menyediakan fungsi-fungsi yang menyediakan fungsi-fungsi baru dan fitur sebagai perluasan dari *StarUML*. Model dapat dibuat sebagai kombinasi dari beberapa elemen-elemen di dalam suatu modul.

2.2.7. C# (C-Sharp)

C# merupakan salah satu aplikasi yang memiliki kemampuan dalam penguatan *Framework.NET*. C# dibuat sejalan dengan perkembangan *Framework.NET*, C# sendiri dikembangkan oleh Microsoft. Dalam penerapannya *C-Sharp* (C#) menjanjikan produktivitas, fleksibilitas serta kemudahan yang ada dari aplikasi sebelumnya yaitu *Visual Basic*, *Java* dan *C++*. C# mengadopsi kemampuan dari penggabungan aplikasi sebelumnya (Yahya, Nur, A. M., 2018).

C# (*C sharp*) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan Microsoft sebagai bagian dari inisiatif kerangka *NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 *C# Language Specification* nama C# terdiri atas sebuah huruf latin C(U+0043) yang diikuti oleh tanda pagar yang menandakan angka #(U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar #

(U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar. C# juga salah satu bahasa pemrograman yang dirancang untuk *Common Language Infrastructure (CLI)*.

2.2.8. Transition

Transition dalam konteks pengembangan aplikasi atau game adalah proses perubahan atau perpindahan dari satu keadaan atau tampilan lainnya. Transisi ini sangat penting untuk memastikan alur navigasi yang lancar dan intuitif bagi pengguna.

Fungsi transition meningkatkan user experience, transisi yang halus dan tepat waktu dapat meningkatkan pengalaman pengguna dengan membuat navigasi menjadi lebih intuitif dan menyenangkan. Memberikan feedback visual, transisi memberikan tanda visual kepada pengguna bahwa tindakan mereka telah diterima dan diproses oleh sistem. Mengatur fokus pengguna, dengan transisi yang baik, pengguna dapat dengan mudah mengalihkan perhatian mereka dari satu bagian lainnya tanpa kebingungan. Memperkuat branding dan estetika, desain transisi yang konsisten dengan tema aplikasi dapat memperkuat identitas visual dan estetika keseluruhan.

Transition adalah elemen dalam desain UI/UX yang membantu meningkatkan kenyamanan dan kepuasan pengguna. Dengan transisi yang dirancang dengan baik, aplikasi atau situs web dapat menawarkan pengalaman yang lebih halus, intuitif, dan estetis, yang pada gilirannya dapat meningkatkan keterlibatan dan retensi pengguna.