

BAB 2

TINJAUAN PUSTAKA

2.1 Profil SD Al-Ihsan Islamic School

Sekolah Dasar Al-Ihsan Islamic School merupakan salah satu Sekolah Dasar yang ada di Provinsi Jawa Barat. Berlokasi di Arcamanik, Jalan Cisaranten Baru, Kecamatan Arcamanik, Kota Bandung, Jawa Barat. Berdiri di bawah naungan Kementerian Pendidikan dan Kebudayaan, SD Al-Ihsan Islamic School didirikan pada tahun 2017.

2.2 Pembelajaran Kosakata Bahasa Arab

Kosakata adalah sekumpulan kata yang dikenal atau digunakan oleh kelompok tertentu .dalam konteks pembelajaran. Kosakata merupakan bagian penting dari Bahasa yang perlu dipelajari untuk meningkatkan keterampilan berbahasa, baik dalam hal pemahaman maupun penggunaan Bahasa yang efektif. Pembelajaran kosakata sangat penting karena menjadi pondasi dasar untuk pengembangan keterampilan Bahasa seperti mendengarkan, berbicara, membaca dan menulis. Ini dapat meningkatkan kemampuan seseorang dalam memahami teks dan komunikasi secara lisan.

Berdasarkan keterangan di atas, maka dapat disimpulkan bahwasanya meningkatkan penguasaan kosakata peserta didik diantaranya dengan menggunakan permainan kata,daftar kata dan kuis kata. Mempelajari Bahasa tidak hanya dengan mempelajari kosakatanya melainkan harus dilakukan dalam belajar Bahasa yaitu dengan menghafal kosakatanya. Pembelajaran kosakata yaitu proses penyampaian bahan pembelajaran yang berupa kata atau untaian kata sebagai unsur dalam pembelajaran Bahasa Arab.

2.3 Game Edukasi

Game edukasi adalah jenis game yang tidak hanya dapat menghibur tetapi juga dapat mengajar karena pengguna dapat bermain sambil belajar. Game, menurut Greg Costikyan, dapat didefinisikan sebagai karya seni di mana pemain, yang disebut pemain, membuat keputusan untuk menggunakan benda-benda di dalam game untuk mengendalikan sumber daya yang dimilikinya untuk mencapai

tujuan tertentu. Permainan didefinisikan oleh Joan Freeman dan Utami Munandar sebagai suatu kegiatan yang membantu anak berkembang secara fisik, intelektual, sosial, moral, dan emosional.

2.4 Game Susun Kata

Game susun kata adalah jenis permainan yang mengharuskan pemain menyusun huruf-huruf acak menjadi kata-kata yang bermakna. Game ini bisa dimainkan secara individu atau dalam kelompok dan sering digunakan sebagai alat untuk meningkatkan kemampuan bahasa, khususnya dalam hal kosakata, ejaan, dan pemahaman kata. Tujuan utama dari game susun kata adalah untuk mengembangkan keterampilan berbahasa pemain, seperti mengenali dan memahami kata-kata dalam suatu bahasa, serta memperkaya kosakata. Selain itu, game ini juga bertujuan untuk melatih logika dan pemikiran kritis, karena pemain harus mampu mengidentifikasi pola dari huruf-huruf yang ada untuk membentuk kata yang benar. Game ini juga dapat meningkatkan kemampuan pemecahan masalah dan kreativitas, karena pemain harus berpikir kreatif untuk menemukan kombinasi huruf yang tepat. Dengan ini pengembang membuat game susunkata sebagai alternatif dalam pembelajaran.

2.5 Game Kuis

Game quiz adalah jenis game interaktif Dimana pemain diuji dengan serangkaian pertanyaan dan diminta untuk memberikan jawaban yang benar. Game kuis seringkali dirancang untuk menguji pengetahuan anak dalam belajar. Game kuis dapat membantu anak dalam meningkatkan pengetahuan termasuk kosakata Bahasa Arab juga pemahaman dalam memperluas kosakata Bahasa Arab. Dengan ini pengembang membuat game kuis dalam pembelajaran kosakata.

2.6 Game Puzzle

Game puzzle adalah jenis game yang dirancang untuk menguji kemampuan pemain dalam memecahkan masalah atau menyelesaikan tantangan tertentu dengan menggunakan logika . Game ini cocok bagi anak anak dalam tahapan pembelajaran karena memberikan anak anak untuk berpikir lebih terampil. Dengan memnafaatkan game puzzle pembelajaran dalam kosakata Bahasa arab dapat

menjadi lebih menarik dan interaktif bagi pengguna. Dengan begitu anak-anak dalam meningkatkan pemahaman kosakata lebih efektif.

2.7 Unity 3D

Unity adalah aplikasi yang digunakan untuk mengembangkan game multi-platform yang didesain untuk mudah digunakan. Unity sangat bagus dan dipadukan dengan aplikasi profesional lainnya. Editor Unity dibuat dengan user interface yang sederhana, setelah ribuan jam pengembangan untuk menjadikannya salah satu editor game terbaik di dunia. Grafis Unity mendukung OpenGL dan DirectX dengan kualitas grafis tingkat tinggi. Unity mendukung berbagai format file, termasuk format umum yang berasal dari aplikasi seni. Aplikasi ini kompatibel dengan versi 64-bit dan dapat beroperasi pada Mac OS X serta Windows, menghasilkan game untuk berbagai platform seperti Mac, Windows, Wii, iPhone, iPad, dan Android.

Selain digunakan untuk membuat video game 3D, Unity juga dapat digunakan untuk membuat animasi 3D real-time, visualisasi arsitektur, dan konten interaktif lainnya. Editor Unity dapat memanfaatkan plugin untuk web player, menghasilkan game browser yang kompatibel dengan Windows dan Mac. Plugin web player juga dapat digunakan untuk widget Mac. Unity bahkan mendukung konsol terbaru seperti PlayStation 3 dan Xbox 360. Pada tahun 2010, Unity memperoleh Technology Innovation Award dari Wall Street Journal, dan pada tahun 2009, Unity Technologies menjadi salah satu dari 5 perusahaan game terbesar. Pada tahun 2006, Unity memenangkan juara kedua pada Apple Design Awards.

Server aset Unity memungkinkan penggunaan semua skrip dan aset game sebagai solusi untuk kontrol versi, mendukung proyek dengan ukuran besar yang berisi ribuan file multi-megabyte. Editor Unity dapat menyimpan metadata dan versi file, menjalankan pembaruan, serta membandingkan versi grafis. Server aset Unity juga kompatibel dengan Mac, Windows, dan Linux, serta menggunakan PostgreSQL sebagai database server open source.

Selain fitur-fitur tersebut, Unity dapat diperluas menggunakan **DOTween**, sebuah library animasi yang memungkinkan developer membuat animasi berbasis tween dengan mudah dan efisien. **DOTween** mendukung berbagai animasi seperti pergerakan objek, perubahan warna, ukuran, rotasi, skala, dan banyak lagi, hanya dengan beberapa baris kode sederhana. Misalnya, developer dapat menggerakkan sebuah objek dalam game dari satu posisi ke posisi lain menggunakan perintah seperti:

```
if(isMid)
{
    characterModel.transform.DOLocalRotate(new Vector3(0,0,-45),0.5f,RotateMode.Fast);
}
```

Gambar 2.1 Contoh source code DOTween

Kode di atas menggerakkan objek ke posisi (0, 10, 0) dalam waktu 2 detik. **DOTween** sangat efisien dan fleksibel, mendukung berbagai jenis easing, loop, serta integrasi yang mudah dengan coroutine dan event Unity, membuatnya ideal untuk mengembangkan animasi dinamis dan interaktif di game.

2.8 Algoritma Fisher Yates-Shuffle

Algoritma *Fisher-Yates Shuffle* adalah sebuah algoritma yang dirancang untuk mengacak elemen-elemen dalam suatu array atau list secara efisien dan tanpa bias. Algoritma ini bekerja dengan cara memilih elemen secara acak dari list yang belum diacak dan menukarnya dengan elemen lain di posisi tertentu, hingga seluruh elemen telah teracak. Algoritma ini memiliki kompleksitas waktu $O(n)$, di mana n adalah jumlah elemen dalam list, sehingga sangat efisien digunakan dalam banyak aplikasi, termasuk dalam pengembangan game.

Dalam pengembangan game, khususnya game kuis, algoritma ini sering digunakan untuk mengacak urutan pertanyaan, jawaban, atau bahkan elemen-elemen grafis, sehingga setiap sesi permainan terasa berbeda. Ini memastikan bahwa pemain tidak akan menghafal urutan elemen dan memberikan pengalaman bermain yang lebih dinamis. Berikut contoh algoritma pengacakan *fisher yates-shuffle*. Sebelum menggunakan algoritma *fisher yates-shuffle* terdapat pada Gambar 2.2.

```
List<string> questions = new List<string> { "Q1", "Q2", "Q3", "Q4", "Q5" };
```

Gambar 2.2 Algoritma *fisher yates shuffle*

Setelah menggunakan metode pengacakan *fisher yates-shuffle*, maka berubah menjadi seperti pada Gambar 2.3.

```
List<string> questions = new List<string> { "Q3", "Q1", "Q5", "Q2", "Q4" };
```

Gambar 2.3 Algoritma *fisher yates-shuffle*

Dalam hal ini algoritma *fisher yates-shuffle* sebagai teknik pengacakan yang efisien untuk mengubah urutan elemen dalam game.

2.9 Android

Android adalah sistem operasi mobile yang sangat populer dan banyak digunakan di berbagai perangkat, mulai dari smartphone hingga Tablet. Dalam konteks pengembangan game edukasi, Android menyediakan platform yang ideal untuk menciptakan aplikasi pembelajaran yang interaktif, mudah diakses, dan efektif. Sebagai sistem operasi yang open-source, Android memungkinkan pengembang untuk memanfaatkan berbagai alat dan teknologi yang dapat disesuaikan dengan kebutuhan spesifik dari game edukasi yang sedang dikembangkan.

Penggunaan Android dalam pengembangan game edukasi memberikan beberapa keunggulan. Pertama, Android mendukung berbagai bahasa pemrograman seperti Java dan Kotlin, yang memungkinkan pengembang untuk menciptakan aplikasi dengan antarmuka pengguna yang intuitif dan fungsionalitas yang canggih. Dengan menggunakan game engine seperti Unity, pengembang dapat merancang game edukasi dengan grafis yang menarik dan gameplay yang interaktif, yang dapat meningkatkan keterlibatan dan motivasi siswa dalam proses pembelajaran.

Selain itu, perangkat Android dilengkapi dengan berbagai sensor dan fitur seperti layar sentuh, akselerometer, dan konektivitas internet, yang dapat dimanfaatkan untuk menciptakan pengalaman belajar yang dinamis dan interaktif. Misalnya, siswa dapat berinteraksi dengan game melalui gestur atau memiringkan

perangkat, atau mereka dapat berkolaborasi dengan siswa lain secara online melalui konektivitas internet yang disediakan oleh Android.

Distribusi game edukasi yang dikembangkan untuk Android juga dipermudah dengan adanya Google Play Store, di mana pengembang dapat mengunggah dan mendistribusikan aplikasi mereka ke jutaan pengguna di seluruh dunia. Ini memungkinkan game edukasi untuk mencapai audiens yang lebih luas dan memberikan dampak yang lebih besar dalam proses pembelajaran.

Dengan berbagai keunggulan tersebut, Android menjadi platform yang sangat mendukung pengembangan game edukasi. Fleksibilitas, aksesibilitas, dan dukungan teknologi yang kuat dari Android memungkinkan pengembang untuk menciptakan game edukasi yang tidak hanya menarik dan menyenangkan, tetapi juga efektif dalam meningkatkan pemahaman dan keterlibatan siswa dalam proses belajar.

2.10 Game

Dalam kamus bahasa Indonesia game adalah permainan. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Permainan merupakan sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga santai. Permainan biasanya dilakukan sendiri atau bersama-sama.[10]

Permainan video (video game) adalah permainan yang menggunakan interaksi antarmuka pengguna melalui Gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan sistem penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan. Sistem elektronik yang digunakan untuk menjalankan permainan video disebut platform, contohnya adalah komputer pribadi dan konsol permainan

2.11 Unified Modeling Language (UML)

Menurut Rosa dan Shalahuddin mendefinisikan UML (Unified Modeling Language) adalah “Salah standar bahasa yang banyak digunakan di dunia industri

untuk mendefinisikan requirement, membuat analisa dan desain[13], serta mengGambarkan arsitektur dalam pemrograman berorientasi objek”. Sedangkan Mulyani mengatakan UML (Unified Modeling Language) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

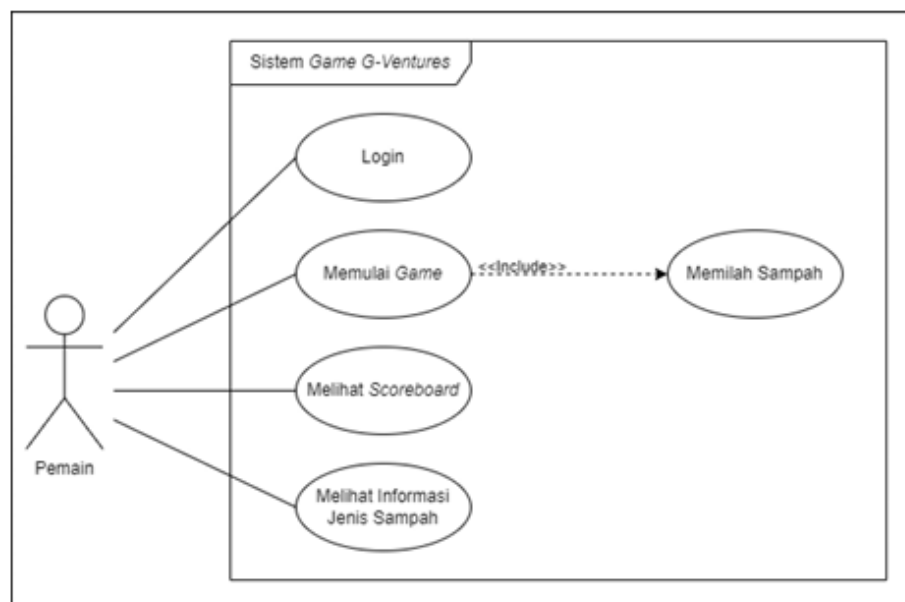
Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (Unified Modeling Language) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek. Ada hal-hal yang sangat mendasar di dalam model UML. Beberapa hal tersebut adalah sebagai berikut:

1. Classes, diuraikan sebagai sekelompok dari objek yang mempunyai atribut, operasi, dan hubungan semantik. Sebuah kelas mengimplementasikan 1 atau lebih interfaces. Sebuah kelas dapat diGambarkan sebagai sebuah persegi panjang, mempunyai sebuah nama, atribut dan metode pengoperasiannya.
2. Interfaces, merupakan sebuah antarmuka yang menghubungkan dan melayani antar kelas dan atau elemen. Interface atau antarmuka mendefinisikan sebuah set atau kelompok dari spesifikasi pengoperasian, umumnya diGambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah atarmuka berdiri sendiri dan umunya merupakan pelengkap dari kelas atau komponen.
3. Collaboration, yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas atau elemen-elemen yang bekerja secara bersama-sama.
4. Use cases, adalah uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Use case digunakan untuk membentuk tingkah laku benda dalam sebuah model serta di realisasikan oleh sebuah collaboration.
5. Nodes, merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalah sebuah komputer,

umumnya mempunyai sedikitnya memori dan prosesor. Sekelompok komponen mungkin terletak pada sebuah node dan juga mungkin akan berpindah dari node satu ke node lainnya. Umumnya node ini digambarkan seperti kubus serta hanya mengandung namanya.

2.12 Usecase Diagram

Use case adalah sebuah konsep dalam rekayasa perangkat lunak yang digunakan untuk mendeskripsikan interaksi antara sistem dan entitas atau aktor yang terlibat dalam suatu skenario tertentu. Dalam landasan teorinya, use case memberikan gambaran tentang bagaimana sistem berperilaku dalam situasi nyata atau skenario penggunaan tertentu. Use case membantu mengidentifikasi fungsionalitas utama sistem dan menyajikan informasi secara terstruktur mengenai langkah-langkah yang harus diambil oleh sistem untuk memenuhi kebutuhan pengguna atau aktor tertentu. Dengan menganalisis use case, pengembang perangkat lunak dapat memahami secara rinci bagaimana interaksi antara pengguna dan sistem harus terjadi, sehingga memudahkan dalam perancangan dan pengembangan sistem yang efisien dan sesuai dengan kebutuhan pengguna. Berikut adalah contoh use case pada Gambar 2.1.



Gambar 2.4 Contoh Usecase Diagram

Berikut ini adalah beberapa komponen utama dalam use case diagram:

1. Sistem digambarkan ke dalam bentuk persegi. Fungsinya untuk membatasi use case dengan interaksi dari luar sistem.
2. Aktor adalah orang atau sistem lain yang berinteraksi dengan sistem yang akan dibuat.
3. Use Case digambarkan dengan elips digunakan untuk interaksi antar aktor dan sistem.
4. *Association* adalah penghubung antara aktor dengan use case.
5. *Generalization* digunakan untuk menunjukkan spesialisasi aktor .
6. *Include* digunakan untuk menunjukkan bahwa salah satu use case merupakan fungsionalitas use case lain.
7. *Extend* digunakan untuk menunjukkan bahwa salah satu use case merupakan tambahan fungsionalitas use case lain jika syarat terpenuhi.

2.13 Usecase Scenario

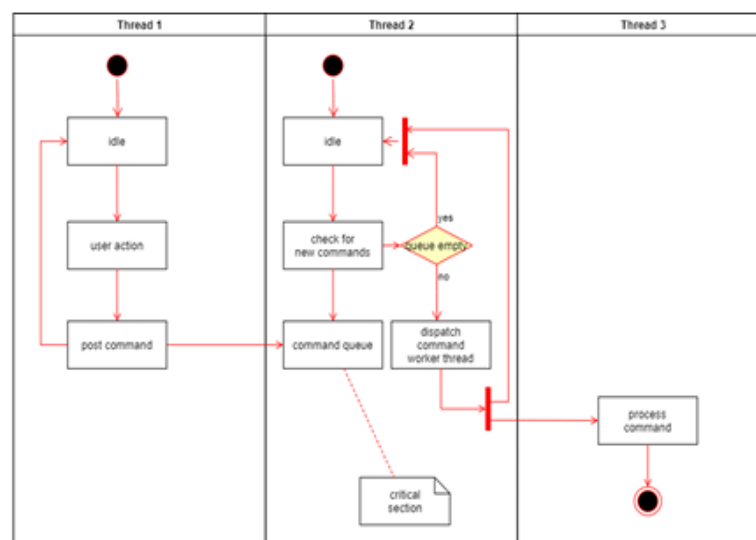
Use Case scenario, dalam konteks pengembangan perangkat lunak, merujuk pada deskripsi lebih lanjut dari suatu use case. Scenario ini memberikan Gambaran yang lebih mendetail tentang bagaimana suatu sistem berinteraksi dengan aktor atau entitas tertentu dalam suatu situasi atau kejadian khusus. Secara sederhana, use case scenario memberikan narasi tentang langkah-langkah atau aksi yang terjadi dalam rangka mencapai tujuan tertentu yang diinginkan oleh pengguna atau aktor sistem. Contoh use case scenario bisa melibatkan urutan langkah-langkah, interaksi antara pengguna dan sistem, serta respons sistem terhadap masukan atau kejadian tertentu. Scenario ini membantu dalam pemahaman lebih lanjut tentang bagaimana sistem berperilaku dalam konteks penggunaan nyata. Berikut adalah contoh use case scenario pada Gambar 2.2.

Use Case	Melihat Info Layanan Akademik
Tujuan	Mahasiswa dapat melihat info terbaru dari BSI pusat
Aktor	User Mahasiswa
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Mahasiswa memilih menu yang tersedia di website. 2. Sistem menampilkan layanan informasi mahasiswa, dan mahasiswa dapat melihat info yang tersedia. 3. Mahasiswa bisa mengakses layanan akademik dan mendownload informasi tersebut.
Skenario Alternatif	1. Jika info mahasiswa yang dipilih tidak tersedia, maka sistem menampilkan pesan “data informasi kosong/masih belum tersedia”. 2. Jika info yang dipilih tidak lengkap, maka sistem akan menampilkan pesan “kosong”.
Kondisi Akhir	Sistem menampilkan layanan informasi yang di pilih mahasiswa dan mahasiswa bisa mencetak informasi tersebut.

Gambar 2.5 Contoh Usecase Scenario

2.14 Activity Diagram

Activity Diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk memodelkan aktivitas, proses, dan alur kerja dalam suatu sistem. Diagram ini membantu dalam menggambarkan serangkaian aktivitas atau tindakan yang terjadi dalam suatu proses atau use case tertentu. Activity diagram memvisualisasikan alur kerja dari suatu kegiatan, menunjukkan urutan aktivitas, pengambilan keputusan, dan percabangan dalam proses. Berikut contoh activity diagram pada Gambar 2.3.



Gambar 2.6 Contoh Activity Diagram

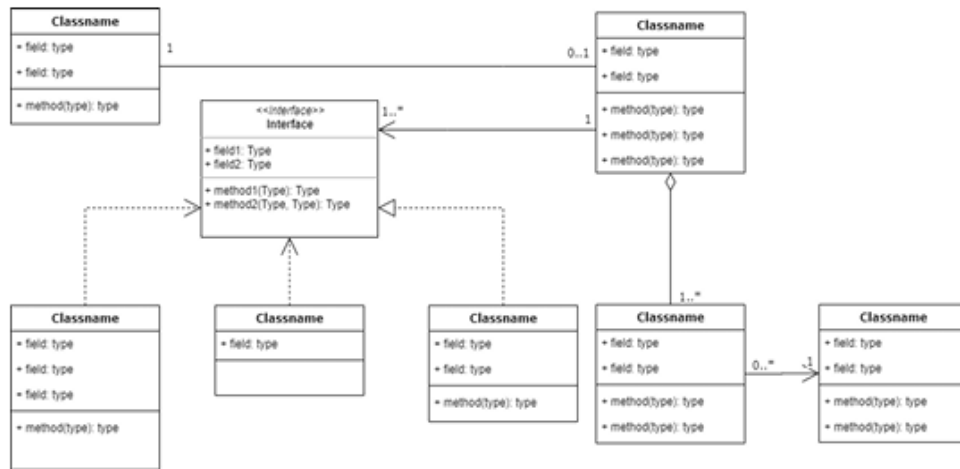
Beberapa elemen penting dalam activity diagram meliputi:

1. Activity: Representasi dari tindakan atau langkah-langkah dalam proses. Biasanya, aktivitas direpresentasikan dengan kotak elips.
2. Flow Edge: Panah yang menghubungkan aktivitas-aktivitas dan menunjukkan urutan atau alur kerja. Garis alur mengindikasikan bagaimana suatu aktivitas mengarah ke aktivitas berikutnya.
3. Decision: Menunjukkan percabangan dalam alur kerja. Pada titik ini, sistem membuat keputusan dan memilih antara beberapa jalur yang berbeda.
4. Merge: Menunjukkan titik di mana beberapa jalur yang berbeda kembali bergabung setelah percabangan.
5. Fork dan Join: Fork menunjukkan titik di mana alur kerja bercabang menjadi beberapa jalur. Join (bergabung) menunjukkan titik di mana jalur-jalur yang berbeda kembali bersatu.
6. Swimlane: Terkadang, diagram aktivitas dapat dibagi menjadi swimlane, yaitu bagian yang menunjukkan pemilik atau pemegang tanggung jawab dari setiap aktivitas.

Activity diagram berguna untuk memahami dan menggambarkan urutan langkah-langkah dalam suatu proses, sehingga membantu dalam analisis dan perancangan sistem dengan lebih jelas.

2.15 Class Diagram

Diagram kelas (class diagram) adalah jenis diagram UML (Unified Modeling Language) yang digunakan dalam rekayasa perangkat lunak untuk memodelkan struktur statis dari suatu sistem, dengan fokus pada kelas dan hubungan antar kelas. Diagram kelas memberikan gambaran tentang entitas-entitas yang ada dalam sistem, atribut-atribut yang dimiliki oleh entitas tersebut, dan hubungan antar entitas. Berikut contoh class diagram pada Gambar 2.4.



Gambar 2.7 Contoh Class Diagram

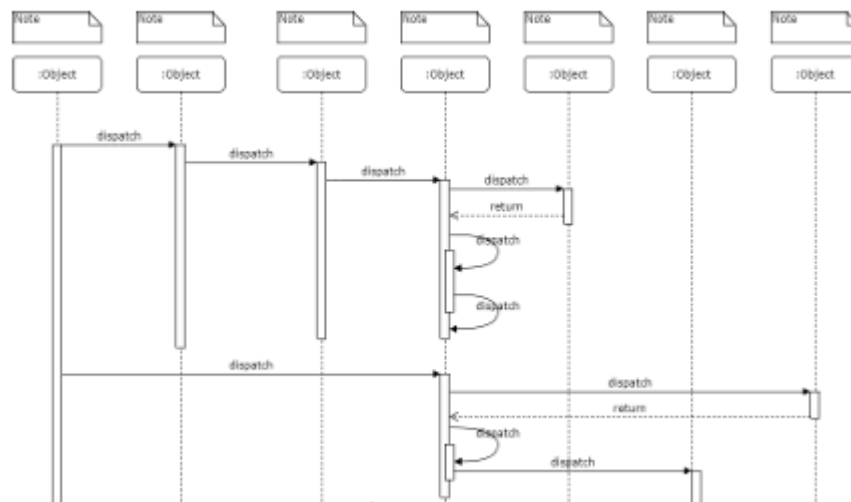
Elemen-elemen utama dalam diagram kelas meliputi:

1. **Class:** Mewakili suatu jenis objek atau entitas dalam sistem. Kelas menggambarkan struktur data dan perilaku yang dimiliki oleh objek tersebut. Kelas direpresentasikan dalam diagram oleh kotak dengan tiga bagian: nama kelas, atribut-atribut, dan metode-metode.
2. **Attribute:** Menunjukkan data atau informasi yang dimiliki oleh suatu kelas. Atribut direpresentasikan oleh nama atribut diikuti oleh tipe data.
3. **Method:** Merupakan tindakan atau fungsi yang dapat dilakukan oleh suatu kelas. Metode direpresentasikan oleh nama metode diikuti oleh parameter dan tipe kembalian (jika ada).
4. **Association:** Menunjukkan hubungan antara dua atau lebih kelas. Asosiasi menggambarkan bagaimana objek dari satu kelas berhubungan dengan objek dari kelas lain.
5. **Agregasi dan Komposisi:** Merupakan bentuk asosiasi yang menunjukkan hubungan bagian-keseluruhan antara kelas-kelas. Agregasi menunjukkan hubungan yang lebih longgar, sedangkan komposisi menunjukkan hubungan yang lebih kuat.
6. **Inheritance:** Menunjukkan hubungan hierarki antara kelas-kelas. Kelas yang mewarisi disebut kelas anak atau subclass, sementara kelas yang memberikan warisan disebut kelas induk atau superclass.

Diagram kelas membantu dalam merancang dan memahami struktur statis dari suatu sistem perangkat lunak. Diagram ini juga digunakan sebagai landasan untuk implementasi kode program dalam berbagai bahasa pemrograman.

2.16 Sequence Diagram

Diagram Sequence (*sequence diagram*) adalah jenis diagram *UML (Unified Modeling Language)* yang digunakan untuk menggambarkan interaksi antara objek-objek dalam suatu skenario atau proses tertentu. Diagram ini menunjukkan urutan pesan atau panggilan metode yang terjadi antara objek-objek selama eksekusi suatu use case atau scenario. Berikut adalah contoh *sequence diagram* pada Gambar 2.5.



Gambar 2.8 Contoh Sequence Diagram

Elemen-elemen utama dalam diagram sequence meliputi:

1. Object: Mewakili instance dari suatu kelas. Objek direpresentasikan dengan sebuah kotak yang berisi nama objek dan garis hidup (lifeline) yang menunjukkan durasi hidup objek selama proses.
2. Message: Menunjukkan panggilan metode atau pertukaran pesan antara objek-objek. Pesan direpresentasikan dengan garis panah yang menghubungkan objek-objek, dan dapat diberi label untuk menunjukkan nama metode yang dipanggil.

3. Lifeline: Menunjukkan durasi hidup objek selama proses. Garis hidup direpresentasikan oleh garis vertikal yang menghubungkan objek dengan garis waktu (time axis).
4. Activation: Menunjukkan periode waktu ketika suatu objek sedang melakukan operasi atau merespon pesan. Aktivasi direpresentasikan oleh kotak yang menggantikan garis hidup dan menunjukkan waktu di mana objek sedang aktif.
5. Interaction Fragment: Bagian dari diagram yang menunjukkan kondisi, percabangan, atau iterasi dalam urutan interaksi. Contoh fragment interaksi adalah fragment opsional (optional) dan loop.

Diagram sequence membantu dalam memodelkan dan memahami bagaimana objek-objek saling berinteraksi dan bertukar pesan selama eksekusi suatu proses atau use case. Diagram ini sering digunakan dalam fase analisis dan desain untuk menggambarkan dinamika eksekusi sistem secara visual.

2.17 Uji Normalitas Gain

Uji Normalitas Gain/n-gain menurut hake adalah sebuah uji yang bisa memberikan Gambaran umum peningkatan skor hasil pembelajaran antara sebelum dan sesudah diterapkannya metode tersebut

Uji gain ternormalisasi (N-Gain) dilakukan untuk mengetahui peningkatan hasil belajar kognitif siswa setelah diberikan perlakuan. Peningkatan ini diambil dari nilai pretest dan posttest yang didapatkan oleh peserta didik. Gain ternormalisasi atau yang disingkat dengan N-Gain merupakan perbandingan skor gain aktual dengan skor gain maksimum. Skor gain aktual yaitu skor gain yang diperoleh peserta didik sedangkan skor gain maksimum yaitu skor gain tertinggi yang mungkin diperoleh peserta didik.

Uji Normalitas Gain/ *Normalized Gain* (N-Gain) menurut hake adalah sebuah uji yang bisa memberikan Gambaran umum peningkatan skor hasil pembelajaran antara sebelum dan sesudah diterapkannya metode tersebut. Berdasarkan definisi gain dan n gain di atas, kita bisa melihat selisih perbedaan skor kemampuan siswa, baik dalam bentuk peningkatan maupun penurunan., sehingga pengujian ini merupakan metode yang cocok untuk

diterapkan untuk menentukan ada tidaknya perkembangan. Perhitungan skor *Normalized Gain* (N-Gain) dapat dinyatakan dalam rumus berikut:

$$g = \frac{Sf - Si}{\text{maximum skor} - Si}$$

Gambar 2.9 Uji Normalitas Gain

Uji gain ternormalisasi (N-Gain) dilakukan untuk mengetahui peningkatan hasil belajar kognitif siswa setelah diberikan perlakuan. Peningkatan ini diambil dari nilai pretest dan posttest yang didapatkan oleh peserta didik. Gain ternormalisasi atau yang disingkat dengan N-Gain merupakan perbandingan skor gain aktual dengan skor gain maksimum. Skor gain aktual yaitu skor gain yang diperoleh peserta didik sedangkan skor gain maksimum yaitu skor gain tertinggi yang mungkin diperoleh peserta didik.