

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Arief, dkk (2019) mengenai Penerapan Metode *Linear Congruent Method* (LCM) Pada Perangkat Lunak Tebak Huruf Hiragana Berbasis *Android*, Arief, dkk menyimpulkan bahwa Aplikasi ini mengimplementasikan metode *linear congruant method* sebagai metode pengacakannya yang berjalan baik pada perangkat android.

Berdasarkan penelitian yang dilakukan oleh Istianto dan Khadafi (2021) mengenai *Random Number Generate* Menggunakan Metode *Linear Congruent Method* (LCM) Untuk Soal Ujian Di SMA Ta'miriyah Surabaya, Istianto dan Khadafi menyimpulkan bahwa Dengan diterapkannya metode LCM pada sistem ini dapat dibangun sebuah aplikasi ujian *online*.

Berdasarkan penelitian yang dilakukan oleh Marsudi dan Rosnelly (2021) mengenai *Random Number Generate* Menggunakan Metode *Linear Congruent Method* (LCM) Untuk Soal Ujian Di SMA Ta'miriyah Surabaya, Istianto dan Khadafi menyimpulkan bahwa Dengan diterapkannya metode LCM pada sistem ini dapat dibangun sebuah aplikasi ujian *online*.

Berdasarkan penelitian yang dilakukan oleh Syiamtoni, dkk (2020) mengenai Perancangan Aplikasi Game Iq Test Dengan Mengimplementasikan *Linear Congruent Method* (LCM), Syiamtoni, dkk menyimpulkan bahwa Aplikasi Game IQ Test ini mengimplementasikan *Linear Congruent Method* sebagai

penyusun sebuah soal atau pertanyaan dimana pengacakan kemunculan soal akan terjadi pengulangan pada waktu tertentu.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal terkait penelitian :

II.2.1. Perancangan

Perancangan merupakan sebuah alur atau proses aplikasi macam-macam teknik prinsip dengan tujuan pendefinisian suatu sistem, perangkat, atau proses detail yang mencukupi untuk memungkinkan pengaktualan fisiknya. (Saputri, 2020: 42).

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Pengertian perancangan menurut para ahli diantaranya adalah:

1. Menurut Varzello / John Reuter III perancangan adalah tahap setelah analisis dari siklus pengembang sistem: Pendefinisian dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi: “Mengembangkan bagaimana suatu sistem dibentuk”.
2. Menurut John Buch & Gary Grudnitski perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
3. Menurut George M. Scott perancangan adalah menentukan bagaimana sistem

akan menyelesaikan apa yang mesti diselesaikan ; tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. (Hidayatulloh, dkk, 2020: 20).

II.2.2. Aplikasi

Aplikasi merupakan penggunaan dalam suatu komputer, instruksi atau pernyataan yang di susun sedemikian rupa sehingga komputer dapat memproses input menjadi output. Aplikasi merupakan perangkat lunak yang dapat menjadi alat bantu untuk mempermudah aktivitas manusia dalam kehidupan setiap hari. Aplikasi memiliki arti suatu Teknik pemrosesan data aplikasi untuk suatu pemecahan masalah yang berdasarkan pada sebuah komputasi yang diharapkan ataupun pemrosesan data yang di harapkan. (Wowiling, dkk, 2020: 508).

Aplikasi merupakan suatu kelompok file (*class, form, report*) yang bertujuan untuk melakukan kegiatan tertentu yang saling berhubungan, misalnya seperti *payroll*, aplikasi *fixed asset*, dan yang lainnya. Menurut Kamus Besar Bahasa Indonesia (KBBI), Aplikasi merupakan suatu sistem yang dirancang untuk mengelola data dengan aturan serta ketentuan tertentu dan menggunakan bahasa pemrograman tertentu. Program aplikasi adalah program komputer yang ditulis menggunakan Bahasa pemrograman dan dipergunakan untuk menyelesaikan suatu masalah dan melakukan pekerjaan kebutuhan pengguna. (Solli dan Sompie, 2021: 536).

II.2.3. Simulasi

Simulasi dapat digunakan untuk menerapkan beberapa pendapat terhadap analisa penempatan dimasa mendatang. Penggunaan simulasi seringkali mengarah pada hasil yang optimal maupun mendekati optimal. Simulasi merupakan proses yang terjadi dalam suatu sistem dengan bantuan perangkat komputer dan dilandasi oleh beberapa asumsi tertentu sehingga sistem tersebut dapat dipelajari secara ilmiah. Simulasi merupakan suatu metode yang mampu memberikan perkiraan sistem yang lebih nyata sesuai kondisi operasional dari kumpulan pekerjaan. (Darnis, dkk, 2020: 139).

II.2.4. UNBK

UNBK merupakan sebuah sistem ujian nasional sekolah berbasis komputer dimana siswa mengerjakan ujian menggunakan komputer yang diharapkan dapat membuat ujian menjadi lebih efisien dan efektif di bandingkan dengan ujian manual yang menggunakan kertas sebagai media ujiannya. UNBK atau Ujian Nasional Berbasis Komputer merupakan sebuah sistem ujian nasional dengan menggunakan kedia komputer. Ujian nasional berbasis komputer berbeda dengan ujian nasional berbasis kertas. Untuk pertama kalinya UNBK diterapkan di tahun 2014 secara online dan hanya diujikan di dua sekolah menengah pertama (SMP), yaitu di SMP Indonesia Singapura dan di SMP Indonesia Kuala Lumpur, Dari penerapan sistem UNBK di kedua sekolah menengah pertama tersebut cukup memuaskan dan dapat memberikan dorongan agar menambah literasi siswa

terhadap Teknologi Informasi dan Komunikasi. Di tahun 2015 diterapkan rintisan Ujian Nasional Berbasis Komputer secara bertahap dengan menambah sebanyak 556 sekolah diantaranya 42 SMP/MTs, 135 SMA/MA, 379 SMK di 29 Provinsi dan Luar Negeri. Setelah itu di tahun 2016 UNBK dilaksanakan dengan menambah sebanyak 4382 sekolah diantaranya 984 SMP/MTs, 1298 SMA/MA, dan 2100 SMK. Total sekolah yang mengikuti UNBK pada tahun 2017 meningkat tajam menjadi 30.577 sekolah yang diantaranya 11.096 SMP/MTs, 9.652 SMA/MA dan 9.829 SMK. Naiknya jumlah sekolah UNBK di tahun 2017 ini selaras dengan kebijakan *resources sharing* yang diterbitkan oleh Kemendikbud yaitu memperbolehkan sekolah yang sarana komputernya masih terbatas untuk dapat menerapkan UNBK pada sekolah lain yang sarana komputernya sudah memadai (Widagdo, dkk, 2019: 206).

II.2.5. Metode *Linear Congruent Method* (LCM)

Metode *Linear Congruent Method* (LCM) merupakan metode pembangkitkan bilangan acak yang banyak digunakan dalam program komputer. LCM diperkenalkan oleh D. H. Lehmer pada tahun 1949. Pembangkitan bilangan acak dengan menerapkan metode LCM dapat dilihat di bawah ini:

$$X_n = (\alpha(X_{n-1}) + c) \bmod m \dots\dots\dots(1)$$

Dimana:

α : konstanta pengali ($a < m$)

c : konstanta pergeseran ($c < m$)

m : konstanta modulus (> 0)

X_n : bilangan acak ke n ($\geq 0, < m$). (Istianto dan Khadafi, 2021: 298).

Contoh Implementasi Linear Congruent Method (LCM):

Implementasi LCM yang diimplementasikan pada aplikasi IQ Test sebagai pengacakan soal. Contoh soal: Peneliti akan membuat percobaan pengacakan soal dalam periode tertentu. Pada contoh soal ini peneliti mempunyai 20 soal (dimulai dari no 0).

Jika:

$$a = 1$$

$$c = 7$$

$$m = 20$$

$$X(0) = 4$$

Penyelesaian:

$$X(0) = 4$$

$$X(1) = (1(4)+7) \bmod 20 = 11$$

$$X(2) = (1(11)+7) \bmod 20 = 18$$

$$X(3) = (1(18)+7) \bmod 20 = 5$$

$$X(4) = (1(5)+7) \bmod 20 = 12$$

$$X(5) = (1(12)+7) \bmod 20 = 19$$

$$X(6) = (1(19)+7) \bmod 20 = 6$$

$$X(7) = (1(6)+7) \bmod 20 = 13$$

$$X(8) = (1(13)+7) \bmod 20 = 0$$

$$X(9) = (1(0)+7) \bmod 20 = 7$$

$$X(10) = (1(7)+7) \bmod 20 = 14$$

$$X(11) = (1(14)+7) \bmod 20 = 1$$

$$X(12) = (1(1)+7) \bmod 20 = 8$$

$$X(13) = (1(8)+7) \bmod 20 = 15$$

$$X(14) = (1(15)+7) \bmod 20 = 2$$

$$X(15) = (1(2)+7) \bmod 20 = 9$$

$$X(16) = (1(9)+7) \bmod 20 = 16$$

$$X(17) = (1(16)+7) \bmod 20 = 3$$

$$X(18) = (1(3)+7) \bmod 20 = 10$$

$$X(19) = (1(10)+7) \bmod 20 = 17$$

Dari penyelesaian di atas diperoleh kemunculan susunan nomor soal yang dibangkitkan dari metode LCM adalah sebagai berikut:

4, 11, 18, 5, 12, 19, 6, 13, 0, 7, 14, 1, 8, 15, 2, 9, 16, 3, 10, 17. (Syiamtoni, dkk, 2020: 331).

II.2.6. Web

Web merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019: 151).

II.2.7. *Hyper Text Markup Language (HTML)*

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti *Microsoft Word*. (Sugijanto, dkk, 2020: 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019: 415).

II.2.8. *Hypertext Preprocessor (PHP)*

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019: 415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script/program* tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020: 2).

II.2.9. MySQL

MySQL adalah *Relational Databases Manajemen System* (RDBMS) yang didistribusikan gratis dibawah lisensi GPL (*General Public Licence*). MySQL sendiri merupakan turunan salah satu konsep utama dalam *database* sejak lama yaitu SQL (*Structured Query Language*), SQL merupakan suatu Bahasa yang dipakai didalam pengambilan data pada *relational database* atau basis data yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan Bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam

query data. (Solli dan Somprie, 2021: 537).

II.2.10. *Unified Modelling Language (UML)*

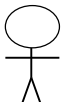
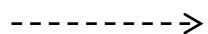
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019: 39).

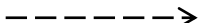
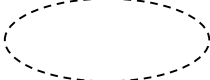
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol Use Case

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber: Andikos, 2019: 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol Activity Diagram

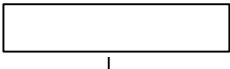
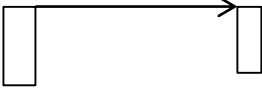
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber: Andikos, 2019: 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol Sequence Diagram

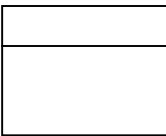
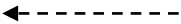
Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

(Sumber: Andikos, 2019: 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber: Andikos, 2019: 39)

