BAB II
LANDASAN TEORI

2.1 Teori-Teori Terkait Bahasan Penelitian (Relevan Theories)

Agar penelitian yang dilakukan dapat terlaksana, maka diperlukan referensi mengenai teori yang terkait dengan penelitian yang akan dilakukan. Teori yang terkait dalam penelitian ini adalah sebagai berikut:

2.1.1 Rancang Bangun

Menurut Kamus Besar Bahasa Indonesia kata rancang berarti mengatur segala sesuatu sebelum bertindak mengerjakan atau melakukan sesuatu untuk merencanakan (Arti Kata Rancang-2 - Kamus Besar Bahasa Indonesia (KBBI) *Online*, n.d.), rancang bangun merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengatruran dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, dengan maksud lain rancang bangun sebuah kegiatan menerjemah hasil analisa ke dalam bentuk paket perangkat lunak kemudian akan menciptakan sistem atau memperbaiki sistem yang ada (Gunawan *et al*., 2021). Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaanya (Gunawan et al., 2021).

Berdasarkan pernyataan di atas dapat disimpulkan bahwa Rancang berarti menyusun sesuatu sebelum dikerjakan. Rancang bangun adalah proses menggabungkan elemen-elemen terpisah menjadi satu sistem yang utuh dan berfungsi. Perancangan merupakan tahap mendefinisikan pekerjaan dengan berbagai teknik, termasuk arsitektur, komponen, dan keterbatasannya.

2.1.2 Aplikasi

Aplikasi berasal dari Bahasa Inggris yaitu *Application* yang artinya penerapan atau penggunaan. Secara teknis *Back-end* aplikasi dibuat oleh para *programmer* atau pengembang dengan menggunakan bahasa pemrograman

tertentu. Aplikasi komputer merupakan sebuah perangkat lunak (*software*) program komputer yang ditulis dalam bahasa pemrograman dan berfungsi melakukan perintah sesuai dengan keinginan dari pembuat aplikasi, Aplikasi komputer dibuat untuk memudahkan pengguna dalam mengerjakan sesuatu menggunakan komputer (Budi Santoso, 2021).

Berdasarkan pernyataan di atas dapat disimpulkan bahwa aplikasi merupakan perangkat lunak yang dikembangkan menggunakan bahasa pemrograman oleh *programmer* guna memudahkan pengguna dalam melaksanakan berbagai tugas atau aktivitas di komputer.

2.1.3 Media Pembelajaran

Media merupakan kata jamak dari kata "*medium*" Kata ini berasal dari bahasa Latin yang berarti antara. Dari sudut pandang komunikasi "*medium*" berarti sesuatu yang dapat menjadi perantara dalam proses komunikasi. "*medium*" dapat juga berarti sesuatu yang dapat membantu penyampaian pesan dan informasi dari sumber pesan (komunikator) kepada penerima pesan (komunikan), Sedangkan media menurut KBBI secara umum merupakan alat (sarana) komunikasi untuk menyampaikan suatu informasi. Adapun media dalam konsep pendidikan ini didefinisikan sebagai alat dan bahan yang digunakan dalam proses pengajaran atau pembelajaran (KBBI Online, 2021).

Berikut ini beberapa pengertian media dari para pakar yang lainnya:

1. Gagne (1970), media adalah berbagai jenis komponen dalam lingkungan anak didik yang dapat memotivasi anak didik untuk belajar.
2. Briggs (1970), media adalah segala alat fisik yang dapat menyajikanpesan serta merangsang anak didik untuk belajar.
3. Arief S. Sadiman (1990), media adalah segala sesuatu yang dapat digunakan untuk menyalurkan pesan dari pengirim kepenerima sehingga dapat merangsang pikiran, perasaan, minat, serta perhatian siswa sedemikian rupa sehingga proses belajar terjadi.

Konsep media pembelajaran harus mengandung dua unsur yakni *software* dan *hardware. Hardware* (perangkat keras) merupakan sarana fisik yang digunakan

dalam kegiatan pembelajaran untuk membantu menyampaikan materi kepada siswa, seperti proyektor, komputer, radio, atau alat peraga lainnya. Sementara itu, *software* (perangkat lunak) adalah konten atau materi pembelajaran yang disajikan melalui perangkat tersebut, misalnya video pembelajaran, modul digital, audio edukatif, atau aplikasi pendidikan. Dengan kata lain, *hardware* berfungsi sebagai media penyampai, sedangkan *software* merupakan isi atau materi yang disampaikan melalui media tersebut (Ginocchio, 2020).

Dari pernyataan di atas dapat disimpulkan bahwa media pembelajaran dalam konteks pendidikan, didefinisikan sebagai alat dan bahan yang digunakan dalam proses pembelajaran. Para ahli mendefinisikan media sebagai segala sesuatu yang dapat memotivasi, merangsang, dan menyalurkan pesan untuk meningkatkan proses belajar. Media pembelajaran terdiri dari dua unsur utama, yaitu *software* (isi atau pesan yang disampaikan) dan *hardware* (alat atau perangkat yang digunakan untuk menyampaikan pesan).

2.1.4 Pramuka

Pramuka dunia didirikan pada tanggal 25 Juli 1907, yang saat itu Robert Baden Powell menjabat sebagai Letnan Jenderal militer Inggris mengadakan perkemahan. Diketahui sejak kecil, Robert Baden Powell menaruh perhatian pada alam dan sangat suka menjelajah hutan yang ada di sekitar sekolahnya (Powell, 2014), di Indonesia sendiri "Gerakan Kepanduan ikut berperan aktif dalam Kongres Pemuda pada 28 Oktober 1928 yang mencetuskan Sumpah Pemuda. Sumpah Pemuda tersebut menjiwai gerakan kepanduan nasional Indonesia pada waktu itu, sehingga kepanduan Indonesia makin berkembang dan menjadi bagian tak terpisahkan dari gerakan perjuangan kemerdekaan Indonesia. K. H. Agus Salim telah mencetuskan idenya dengan mengganti istilah Padvinder menjadi Pandu. Semasa penjajahan Jepang, gerakan kepanduan dilarang, namun idealisme dan semangat kepanduan tetap menjiwai para pandu dalam perjuangan bangsa Indonesia mencapai kemerdekaan para anggota pandu terjun dengan para pemuda merebut kemerdekaan dan mempertahankannya Munculnya istilah Pandu menggantikan istilah *Padvinder* merupakan sebuah semangat perjuangan lepas dari

belenggu penjajahan melalui kegiatan kepemudaan, yang memang pemuda merupakan salah satu penggerak perjuangan merebut kemerdekaan, salah satunya terwujud melalui peristiwa Sumpah Pemuda 1928 dan lahirlah bangsa Indonesia. Menarik dikaji tentang Sumpah Pemuda selain ikrar pengakuan satu tumpah darah, satu bangsa, dan satu bahasa persatuan Indonesia; dalam putusan Kongres Pemuda Indonesia tahun 1928 bahwa persatuan Indonesia diperkuat dengan memperhatikan kemauan, sejarah, hukum adat, pendidikan dan kepanduan. Dengan demikian menunjukkan bahwa pada dasarnya pendidikan dan kepanduan tidak lepas dari sejarah perjuangan bangsa Indonesia(Cinde Adia Diningsih, 2024).

Pada 14 Agustus 1961, Gerakan Pramuka diperkenalkan secara resmi kepada masyarakat luas dalam suatu upacara di halaman Istana Negara. Ditandai dengan penyerahan Panji Gerakan Pramuka dari Presiden Soekarno kepada Sri Sultan Hamengku Buwono IX yang juga menjadi Ketua pertama Kwartir Nasional Gerakan Pramuka. Panji itu lalu diteruskan Sri Sultan Hamengku Buwono IX kepada suatu barisan defile yang terdiri dari para Pramuka di Jakarta, dan dibawa berkeliling kota. Tanggal 14 Agustus itulah yang kemudian ditetapkan sebagai Hari Pramuka dari dirayakan seluruh Pramuka setiap tahunnya (Powell, 2014).

Di dalam dunia pendidikan, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) memastikan Pramuka sebagai ekstrakurikuler wajib disediakan oleh satuan pendidikan. Kepala Badan Standar, Kurikulum, dan Asesmen Pendidikan (BSKAP), Anindito Aditomo menegaskan bahwa setiap sekolah hingga jenjang pendidikan menengah wajib menyediakan Pramuka sebagai kegiatan ekstrakurikuler dalam Kurikulum Merdeka. Peraturan Mendikbudristek Nomor 12 Tahun 2024 tentang Kurikulum pada Pendidikan Anak Usia Dini, Jenjang Pendidikan Dasar, dan Jenjang Pendidikan Menengah mewajibkan sekolah menyelenggarakan minimal satu ekstrakurikuler. Undang-undang Nomor 12 Tahun 2010 tentang Gerakan Pramuka juga mewajibkan satuan pendidikan untuk memiliki gugus depan. “Permendikbudristek 12/2024 tidak mengubah ketentuan bahwa Pramuka adalah ekstrakurikuler yang wajib disediakan sekolah. Sekolah tetap wajib menyediakan setidaknya satu kegiatan ekstrakurikuler, yaitu Pramuka (KEMENDIKBUD RI, 2020).

2.1.5 Sandi Morse

Kode Morse adalah sistem representasi huruf, angka, dan tanda baca dengan menggunakan sinyal kode, Kode Morse diciptakan oleh Samuel F.B. Morse dan Alfred Vail pada tahun 1835. Kode Morse juga digunakan dan dipelajari di dunia kepramukaan atau kepanduan, Dalam dunia kepramukaan kode Morse disampaikan menggunakan senter atau peluit pramuka. Kode Morse disampaikan dengan cara meniup peluit dengan durasi pendek untuk mewakili titik dan meniup peluit dengan durasi panjang untuk mewakili garis (Badaruddin et al., 2020)

Durasi pengiriman kode morse diukur dalam satuan Kata Per Menit (*word per minute*; disingkat WPM), dan berkisar di antara 8-50 WPM di dalam penggunaannya secara umum melalui jaringan radio atau media lain (Badaruddin et al., 2020).

Titik = 1 satuan waktu

Garis = 3 Satuan waktu

Jeda antar huruf = 3 satuan waktu

Jeda antar kata = 7 satuan waktu.

Berikut adalah bentuk Kode Morse :

Alfabet dalam kode morse

A	= •–	J	= •–––	S	= •••	Tanda Baca :	
B	= –•••	K	= –•–	T	= –	Titik (.)	= ••••••
C	= –•–•	L	= •–••	U	= ••–	Titik Koma (;)	= –•–•–•
D	= –••	M	= ––	V	= •••–	Koma (,)	= •–•–•–
E	= •	N	= –•	W	= •––	Titik Dua (:)	= –––•••
F	= ••–•	O	= –––	X	= –••–	Garis Pemisah (-)	= –••••–
G	= ––•	P	= •––•	Y	= –•––	Tanda Seru (!)	= ––•••––
H	= ••••	Q	= ––•–	Z	= ––••	Tanda Tanya (?)	= ••––••
I	= ••	R	=•–•				

Gambar 2. 1 Alfabet Sandi Morse (Badaruddin et al., 2020).

Angka :

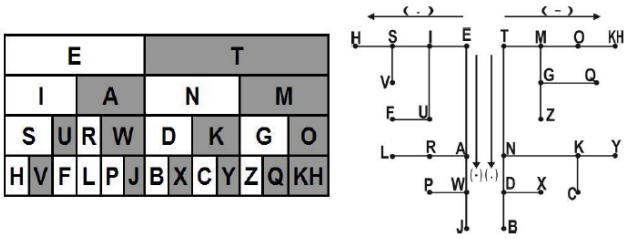
1 = •– – – –	6 = – • • • •
2 = • •– – –	7 = – – • • •
3 = • • •– –	8 = – – – • •
4 = • • • •–	9 = – – – – •
5 = • • • • •	10 = – – – – –

Gambar 2. 2 Angka Sandi Morse (Badaruddin et al., 2020)

2.1.5.1 Metode Penghafalan

Beberapa metode umum digunakan untuk memudahkan penghafalan kode ini, baik *visual*, *auditori* dan metode lain yang masih terus berkembang.

1. Metode rumus bangunan Morse



Gambar 2. 3 Rumus bangunan Morse (Badaruddin et al., 2020).

2. Metode *Koch*

Metode *Koch* adalah metode pembelajaran pengiriman kode morse dengan sistem *gradual*. Latihan dengan metode *Koch* dimulai dengan menggunakan dua huruf yang diulang terus menerus, (umumnya E dan T untuk alasan pembiasaan dengan *interval*)(Badaruddin et al., 2020).

Setelah seseorang menguasai dua huruf ini dan dapat membaca maupun mengirimkannya dengan cepat, maka satu huruf ditambahkan, dan seterusnya hingga seseorang yang mempelajari kode morse dapat menguasai pembacaan maupun pengiriman kode melalui pembiasaan (Badaruddin et al., 2020).

3. Metode Substitusi

Metode ini umum digunakan di kepramukaan Indonesia, yaitu dengan membuat padanan kata yang berawal dari alfabet latin, dan setiap O mewakili garis (-), dan setiap huruf vokal lain mewakili titik (.)

A : Ano (. –)	N : Notes (- .)
B : Bonaparte (- ...)	O : Omoto (- – -)
C : Coba – coba (- . – .)	P : Pertolongan (. – - .)
D : Dominan (- . .)	Q : Qomokaro (- – . –)
E : Egg (.)	R : Rasove (. – .)
F : Father Joe (. . – .)	S : Sahara (. . .)
G : Golongan (- – .)	T : Ton (-)
H : Himalaya (. . . .)	U : U'nesco (. . –)
I : Islam (. .)	V : Versikaro (. . . –)
J : Jago loro (. – - -)	W : Winoto (. – -)
K : Komando (- . –)	X : Xosendero (- . . –)
L : Lemonade (. – . .)	Y : Yosimoto (- . – -)
M : Motor (- -)	Z : Zoroaster (– – ..)

Gambar 2. 4 Metode Subtitusi (Badaruddin et al., 2020).

4. Metode Pengelompokan

Metode yang mengelompokkan huruf-huruf berdasarkan bagaimana huruf ini diwakili oleh kode morsenya. Pengelompokan tersebut antara lain *Alphabet* dengan kode morse yang berkebalikan antara titik dan garis, seperti :

Alfabet dengan kode morse yang berkebalikan :

E	= .	T	= -
I	= ..	M	= --
S	= ...	O	= ---
H	=	KH	= ----
K	= -.-	R	= .-.
X	= -..-	P	= .--.

Alphabet dengan kode morse yang berlawanan

A	= .-	N	= -.
U	= ..-	D	= -..
V	= ...-	B	= -...
W	= .--	G	= --.
F	= ..-.	L	= .-..
Y	= -.--	Q	= --.-

Gambar 2. 5 Metode Pengelompokan (Badaruddin et al., 2020).

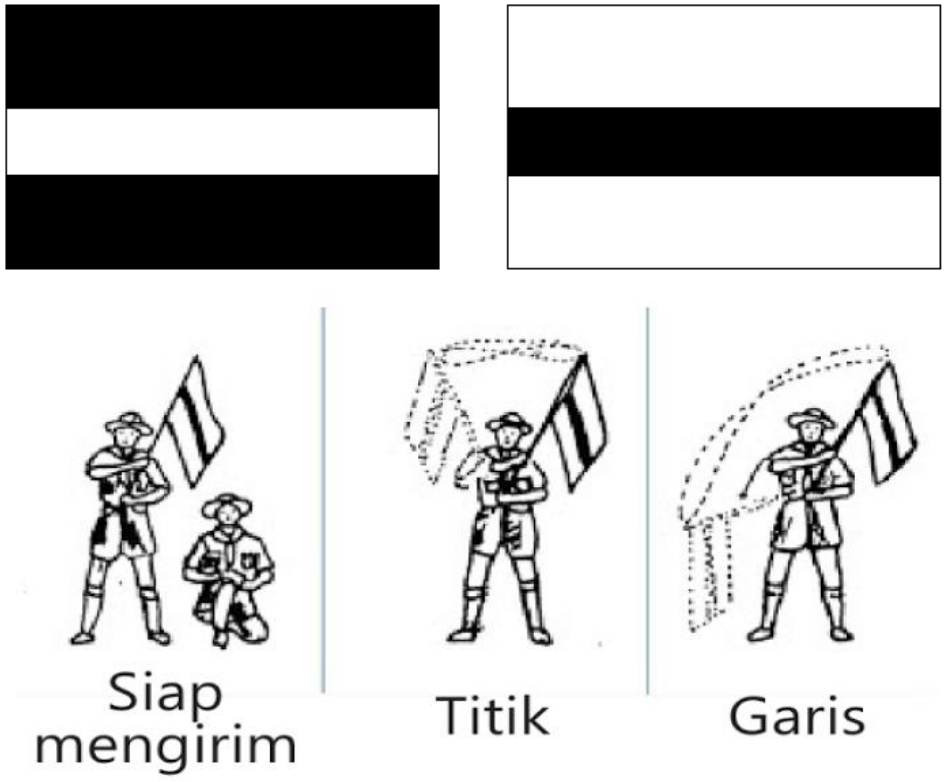
5. Alat - Alat yang Digunakan Dalam Kode Morse

Selain bunyi peluit pendek – panjang, kode morse juga bisa disampaikan dengan menggunakan alat-alat lain, yaitu: (Badaruddin et al., 2020) :

A. Cahaya atau Sinar, seperti : lampu, lilin, senter, api dll
B. Gerak, Seperti : Asap, Bendera, Lambaian tangan, kedipan mata dll.
C. Tulisan, seperti : Sandi, Kode dll
D. Denyut Listrik (pada kabel telegraph)

6. Bendera Morse

Ada dua macam contoh bendera morse. Warna dipilih sesuai dengan latar belakang tempat pengirim, agar bendera dapat dilihat jelas dari jarak jauh.



Gambar 2. 6 Bendera Morse (Badaruddin et al., 2020).

2.1.6 Algoritma

2.1.6.1 Pengertian Algoritma

Kata *algorism* diambil dari nama seorang penulis terkenal dari bangsa Arab yakni Abu Ja'far Muhammad Ibnu Musa al-Khuwarizmi. Ia menciptakan sebuah buku yang diberi judul kitab *aljabar wal-muqabala,* dengan arti " Buku Pemugaran dan Pengurangan" (*The book of restoration and reduction*). Kemudian ditemukanlah akar kata "Aljabar"(*algebra*) pada judul buku tersebut. Karena perhitungan dengan angka Arab semakin populer pada masa itu maka lambat laun istilah tersebut diserap oleh bahasa Indonesia dan menjadikannya algoritma (Maulana Hasan & Yahfizham, 2023).

Pengertian algoritma berdasarkan *Microsoft Press Computer and Internet Dictionary* (1998) merupakan deretan dari sekumpulan rangkaian tindakan tertentu yang pastinya logis dalam menyelesaikan persoalan. Logis di sini memiliki arti bahwa tindakan yang dilakukan harus sesuai dengan urutan dan tersusun secara runtut serta langkah tersebut harus diketahui dengan pasti agar algoritma yang dipakai dapat berjalan semestinya (Maulana Hasan & Yahfizham, 2023).

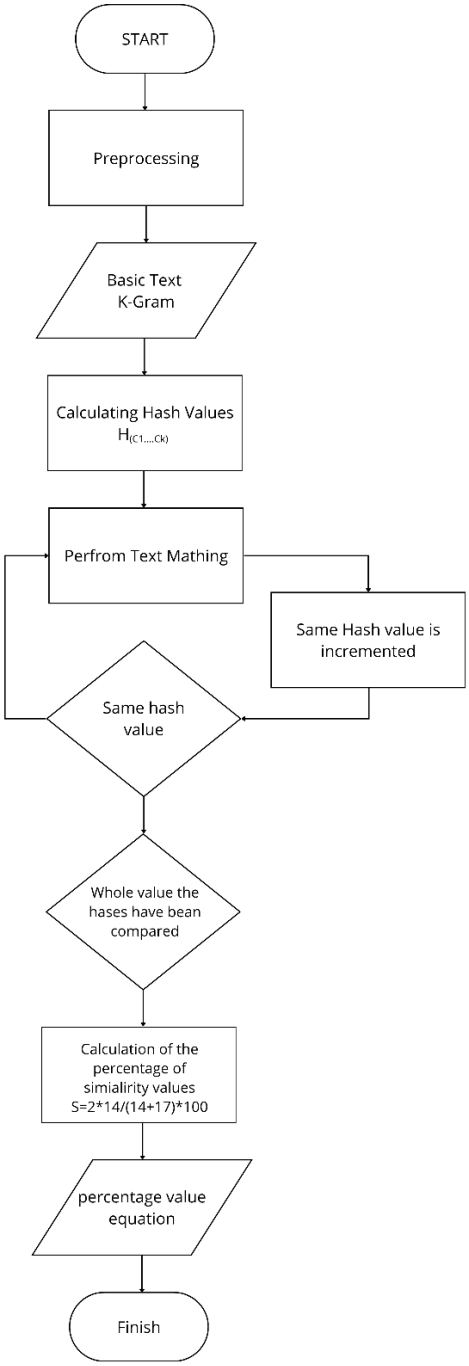
Menurut Sjukani (2005) berpandangan bahwa algoritma merupakan jalan pikiran yang digunakan saat penyelesaian sebuah operasi yang dijabarkan dalam bentuk tertulis. Alur pikir disini mengisyaratkan bahwa jalan pikiran seseorang dapat berbeda dengan yang lainnya. Sementara itu makna tertulis berbentuk kalimat, tabel, atupun gambar tertentu yang dapat ditulis (Maulana Hasan & Yahfizham, 2023).

Dalam pandangan Gunadarma (1988) definisi algoritma merupakan gabungan dari sekumpulan perintah yang menjabarkan langkah-langkah yang jelas saat prosedur pelaksanaan memecahkan sebuah permasalahan, dengan tuntutan himpunan bersifat mekanik (Maulana Hasan & Yahfizham, 2023).

Berdasarkan penjelasan di atas Algoritma merupakan konsep *fundamental* dalam penyelesaian masalah yang berasal dari pemikiran ilmuwan Arab, Al-Khuwarizmi, yang juga menjadi akar kata dari istilah "aljabar" dan "algoritma". Secara umum, algoritma dapat dipahami sebagai sekumpulan langkah logis, terstruktur, dan runtut yang dirancang untuk menyelesaikan suatu permasalahan. Beberapa ahli mendefinisikannya sebagai rangkaian tindakan (*Microsoft Press*), jalan pikiran yang tertulis dalam bentuk kalimat, tabel, atau gambar (Sjukani), serta himpunan perintah mekanik yang menjelaskan langkah-langkah pemecahan masalah (Gunadarma). Meskipun definisinya bervariasi, seluruh pandangan tersebut menekankan pentingnya kejelasan, keterurutan, dan logika dalam penyusunan algoritma.

2.1.6.2 Algoritma Rabin Karp

Algoritma Rabin Karp adalah algoritma pencocokan *string* yang menggunakan fungsi *hash* sebagai pembanding antara *string* yang dicari (m) dengan *substring* pada teks (n). Apabila *hash value* keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya,



Gambar 2. 7 *Flowchart* Algoritma Rabin Karp (Santoso & Solichin, 2023).

Algoritma Rabin Karp ditemukan oleh Michael O. Rabin dan Richard M. Karp. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian

jamak (Putra et al., 2019). Algoritma Rabin Karp memiliki beberapa tahapan yaitu :

A. Preprocessing

Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang mengalami pengolahan lebih lanjut (Putra et al., 2019). Pada *preprocessing* terdapat beberapa tahap, yaitu:

a. Filtering

Filetring adalah tahap mengambil kata – kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words.* Contoh *stopwords* adalah "yang" "dan", "di","dari" dan seterusnya (Putra et al., 2019).

b. Case Folding

Case folding adalah merupakan tahapan yang mengubah semua huruf dalam dokumen atau isi konten yang terdapat pada suatu web menjadi huruf kecil. Hanya huruf „a" sampai dengan „z" yang diterima. Karakter selain huruf dihilangkan dan akan dilakukan proses penghapusan. Pada penelitian ini *case folding* yang diterima adalah huruf "a" sampai dengan "z",selain itu seperti tanda hubung atau tanda baca akan dihapus dandihilangkan (Putra et al., 2019).

c. Parsing K-Gram

K-gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode *K-grams* ini digunakan untuk mengambil potonganpotongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen (Putra et al., 2019).

Berikut ini adalah contoh *K-grams* dengan k=5 :

Teks : “Sandi Morse Pramuka” kemudian dilakukan penghilangan spasi “sandimorsepramuka” sehingga dihasilkan rangkaian 5-*grams* yang diturunkan dari teks : “sandi andim ndimo dimor imors morse orsep rsepr sepra epram pramu ramuk amuka”

d. Hashing

Hashing adalah suatu cara untuk merubah sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut (Putra et al., 2019).

Contoh sederhana *hashing* adalah :

Tabel 2. 1 Contoh perhitungan *Hashing* (Manurung et al., 2022).

Char	**Nilai ASCII (decimal)**	**Char ASCII * basis ^ posisi karakter dimulai dari belakang**
a	97	(97 * (7^2)) +
r	114	(114 * (7^1)) +
o	111	(111 * (7^0))
Hasil hashing [aro]		**= 5662**

e. Rolling Hash

Algoritma Rabin-Karp menggunakan fungsi *hash* yang disebut dengan *rolling hash* untuk menentukan apakah kata-kata yang dicocokkan sama. *Rolling hash* adalah sebuah fungsi *hash* yang input-nya dikelompokkan ke dalam suatu blok yang digerakkan melewati input secara keseluruhan. Beberapa fungsi *hash* memungkinkan *rolling hash* untuk dikomputasi dengan cepat. Nilai *hash* yang baru dapat dengan cepat dihitung dari nilai *hash* yang lama dengan cara menhilangkan nilai lama dari kelompok *hash* dan menambahkan nilai baru ke dalam kelompok tersebut (Putra et al., 2019).

Proses dilakukan dengan menggunakan bantuan kode strlen, ord an pow untuk mentransformasikan sebuah *string* menjadi suatu

nilai yang unik dengan panjang tertentu yang berfungsi sebagai penanda string.

Fungsi *hash* $H_{(C1....Ck)}$ didefinisikan sebagai berikut:

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + ... + c_{(k-1)} * b^{(k)} + c_k$$

Keterangan :

C = Nilai ASCII Karakter

B = Basis Bilangan (B=10 Tanpa Modulus)

K = Banyak Karakter (K=4)

Tabel 2. 2 NIlai ASCII (Santoso & Solichin, 2023).

Dec	***Char***	***Description***	***Dec***	***Char***	***Description***
0	*NUL*	*Null character*	*64*	*@*	*At sign*
1	*SOH*	*Start of Header*	*65*	*A*	*Uppercase A*
2	*STX*	*Start of Text*	*66*	*B*	*Uppercase B*
3	*ETX*	*End of Text*	*67*	*C*	*Uppercase C*
4	*EOT*	*End of Transmission*	*68*	*D*	*Uppercase D*
5	*ENQ*	*Enquiry*	*69*	*E*	*Uppercase E*
6	*ACK*	*Acknowledge*	*70*	*F*	*Uppercase F*
7	*BEL*	*Bell*	*71*	*G*	*Uppercase G*
8	*BS*	*Backspace*	*72*	*H*	*Uppercase H*
9	*TAB*	*Horizontal Tab*	*73*	*I*	*Uppercase I*
10	*LF*	*Line Feed (New Line)*	*74*	*J*	*Uppercase J*
11	*VT*	*Vertical Tab*	*75*	*K*	*Uppercase K*
12	*FF*	*Form Feed*	*76*	*L*	*Uppercase L*
13	*CR*	*Carriage Return*	*77*	*M*	*Uppercase M*
14	*SO*	*Shift Out*	*78*	*N*	*Uppercase N*
15	*SI*	*Shift In*	*79*	*O*	*Uppercase O*
16	*DLE*	*Data Link Escape*	*80*	*P*	*Uppercase P*
17	*DC1*	*Device Control 1*	*81*	*Q*	*Uppercase Q*
18	*DC2*	*Device Control 2*	*82*	*R*	*Uppercase R*
19	*DC3*	*Device Control 3*	*83*	*S*	*Uppercase S*
20	*DC4*	*Device Control 4*	*84*	*T*	*Uppercase T*
21	*NAK*	*Negative Acknowledge*	*85*	*U*	*Uppercase U*
22	*SYN*	*Synchronous Idle*	*86*	*V*	*Uppercase V*
23	*ETB*	*End of Transmit Block*	*87*	*W*	*Uppercase W*

Dec	***Char***	***Description***	***Dec***	***Char***	***Description***
24	*CAN*	*Cancel*	*88*	*X*	*Uppercase X*
25	*EM*	*End of Medium*	*89*	*Y*	*Uppercase Y*
26	*SUB*	*Substitute*	*90*	*Z*	*Uppercase Z*
27	*ESC*	*Escape*	*91*	*[*	*Left Square Bracket*
28	*FS*	*File Separator*	*92*	**	*Backslash*
29	*GS*	*Group Separator*	*93*	*]*	*Right Square Bracket*
30	*RS*	*Record Separator*	*94*	^	*Caret*
31	*US*	*Unit Separator*	*95*	_	*Underscore*
32	*(space)*	*Space*	*96*	`	*Grave Accent*
33	*!*	*Exclamation Mark*	*97*	*a*	*Lowercase a*
34	*"*	*Double Quote*	*98*	*b*	*Lowercase b*
35	*#*	*Number Sign*	*99*	*c*	*Lowercase c*
36	*$*	*Dollar Sign*	*100*	*d*	*Lowercase d*
37	*%*	*Percent Sign*	*101*	*e*	*Lowercase e*
38	*&*	*Ampersand*	*102*	*f*	*Lowercase f*
39	*'*	*Single Quote*	*103*	*g*	*Lowercase g*
40	*(*	*Left Parenthesis*	*104*	*h*	*Lowercase h*
41	*)*	*Right Parenthesis*	*105*	*i*	*Lowercase i*
42	*	*Asterisk*	*106*	*j*	*Lowercase j*
43	+	*Plus Sign*	*107*	*k*	*Lowercase k*
44	*,*	*Comma*	*108*	*l*	*Lowercase l*
45	*-*	*Hyphen-Minus*	*109*	*m*	*Lowercase m*
46	*.*	*Period*	*110*	*n*	*Lowercase n*
47	*/*	*Slash*	*111*	*o*	*Lowercase o*
48	*0*	*Digit 0*	*112*	*p*	*Lowercase p*
49	*1*	*Digit 1*	*113*	*q*	*Lowercase q*
50	*2*	*Digit 2*	*114*	*r*	*Lowercase r*
51	*3*	*Digit 3*	*115*	*s*	*Lowercase s*
52	*4*	*Digit 4*	*116*	*t*	*Lowercase t*
53	*5*	*Digit 5*	*117*	*u*	*Lowercase u*
54	*6*	*Digit 6*	*118*	*v*	*Lowercase v*
55	*7*	*Digit 7*	*119*	*w*	*Lowercase w*
56	*8*	*Digit 8*	*120*	*x*	*Lowercase x*
57	*9*	*Digit 9*	*121*	*y*	*Lowercase y*
58	*:*	*Colon*	*122*	*z*	*Lowercase z*
59	*;*	*Semicolon*	*123*	*{*	*Left Curly Bracket*
60	<	*Less Than*	*124*	|	|
61	=	*Equals Sign*	*125*	*}*	*Right Curly Bracket*

Dec	*Char*	*Description*	*Dec*	*Char*	*Description*
62	*>*	*Greater Than*	*126*	*~*	*Tilde*
63	*?*	*Question Mark*	*127*	*DEL*	*Delete*

Tabel 2. 3 Tahap menghitung jumlah *Hasing* (Manurung et al., 2022).

Pattern	***Speech***	**Jumlah Hash yang Sama**	**Hasil *Similarity***
1. aro=5662.0 2. roa=6460.0 3. oai=6223.0 4. ait=5604.0 5. ita=6054.0 6. tal=6471.0 7. all=5617.0 8. lla=6145.0 9. lad=6071.0 10. adz=5575.0 11. dzi=5859.0 12. ziy=6834.0 13. iyu=6109.0 14. yuk=6855.0 15. uka=6579.0 16. kad=6022.0 17. adz=5575.0 18. dzi=5859.0 19. zib=6811.0 20. ibu=5948.0 21. bub=5719.0 22. ubi=6524.0 23. bid=5637.0 24. idd=5945.0 25. ddi=5705.0 26. din=5745.0	1. aro=5662.0 2. roa=6460.0 3. oai=6223.0 4. ait=5604.0 5. ita=6054.0 6. tal=6471.0 7. all=5617.0 8. lla=6145.0 9. lad=6071.0 10. adz=5575.0 11. dzi=5859.0 12. ziy=6834.0 13. iyu=6109.0 14. yuk=6855.0 15. uka=6579.0 16. kad=6022.0 17. adz=5575.0 18. dzi=5859.0 19. zib=6811.0 20. ibu=5948.0 21. bub=5719.0 22. ubi=6524.0 23. bid=5637.0 24. idd=5945.0 25. ddi=5705.0 26. din=5745.0	26	100%

B. Pencocokan

Pada processing, setiap m deret (*kontigu*) karakter pada *field* pencarian dicari *hash key* dengan cara yang sama dengan pencarian *hash key* untuk *pattern*. Perlu dingat dengan menggunakan aturan *horner*, pencarian untuk m deret karakter deret karakter dapat dengan mudah

dihitung dari predesesornya. Sedangkan untuk proses pencocokannya, dipakai sebuah *teorema* yaitu:

Sebuah *string* A identik dengan *string* B, jika (syarat perlu) *string* A memliki *hash key* yang sama dengan *hash key* yang dimiliki oleh *string*

Contoh :

Tabel 2. 4 Rolling Hash (Manurung et al., 2022).

No Ayat	Pattern	Speech
1	[aro, roa, oai, ait, ita, tal, all, lla, lad, adz, dzi, ziy, iyu, yuk, uka, kad, adz, dzi, zib, ibu, bub, ubi, bid, idd, ddi, din]	[aro, roa, oai, ait, ita, tal, all, lla, lad, adz, dzi, ziy, iyu, yuk, uka, kad, adz, dzi, zib, ibu, bub, ubi, bid, idd, ddi, din]
2	[fad, adz, dza, zal, ali, lik, ika, kal, all, lla, lad, adz, dzi, zii, iiy, iya, yad, adu, duu, uul,	[fad, adz, dza, zal, ali, lik, ika, kal, all, lla, lad, adz, dzi, zii, iiy, iya, yad, adu, duu, uul,

C. Pengukuran Nilai Kemiripan

Pada ahap string matching dilakukan setelah mendapatkan nilai *hash*. Tahap string matching yaitu tahap melakukan perhitungan nilai yang sama antara kunci jawaban dan jawaban mahasiswa dengan menggunakan *Dice's Similarity Coefisient* dengan rumus sebagai berikut:

$$S=2C/(a+b)*100$$

Keterangan :

S= nilai similarity antar kedua teks

A= jumlah dari kumpulan K-grams dalam teks 1

B= jumlah dari kumpulan K-grams dalam teks 2

C= jumlah dari K-grams yang sama dari tek s yang dibandingkan.

Jadi :

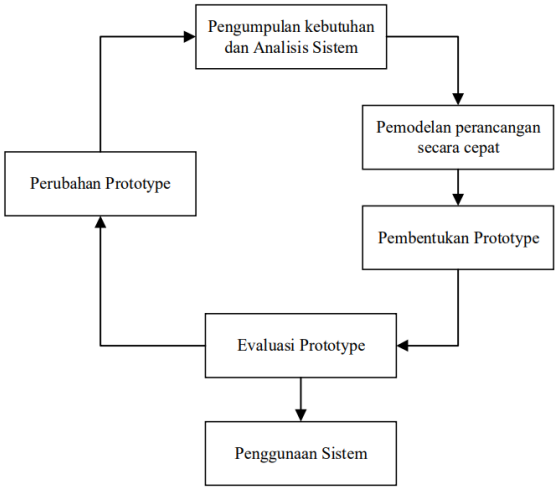
S=2*14/(14+17)*100

S=28/31*100

S=90,32

2.1.7 Prototype

Metode pengembanan perangkat lunak yang digunakan dalam perancangan aplikasi ini adalah dengan menggunakan metode *prototype.* Prototype adalah versi awal dari sebuah sistem perangkat lunak *(software)* yang digunakan untuk mendemonstrasikan konsep-konsep, percobaan rancangan, dan menemukan lebih banyak masalah dan solusi yang memungkinkan. Sistem yang dirancang menggunakan *prototype* memperbolehkan *user* untuk mengetahui bagaimana sistem dapat berjalan dengan baik (Putra, Werthi, & Wijaya, 2020). Tahapan metode *prototyping* yang akan digunakan adalah sebagai berikut (Renaningtias & Apriliani, 2021).



Gambar 2. 8 Model Pengembangan Perangkat Lunak *Prototyping* (Renaningtias & Apriliani, 2021).

1. Pengumpulan Kebutuhan dan Analisis Sistem

Tahapan awal yang digunakan dalam penelitian ini yaitu dengan mengumpulkan kebutuhan dan melakukan analisis sistem. Pada tahapan ini dilakukan identifikasi kebutuhan sistem dan garis besar dari sistem yang akan dibuat. Peneliti melakukan identifikasi permasalahan pada media penyuluhan

kesehatan di Puskesmas Kecamatan Kuningan.

2. Pemodelan Perancangan Secara Cepat

Tahapan selanjutnya yaitu pemodelan perancangan secara cepat yang digunakan sebagai acuan dalam pembuatan model *prototype.*

3. Pembentukan *Prototype*

Dalam tahapan ini, dilakukan pembentukan *prototype* berdasarkan rancangan pemodelan yang telah dilakukan sebelumnya

4. Evaluasi *Prototype*

Pada tahap ini, dilakukan evaluasi terhadap prototype yang disesuaikan dengan kebutuhan. Jika belum sesuai dengan kebutuhan, maka dapat dilakukan tahap selanjutnya yaitu dengan melakukan perubahan pada *prototype.*

5. Perubahan *Prototype*

Tahapan ini dilakukan untuk menyempurnakan *prototype* yang akan dibagun agar menghasilkan *prototype* yang sesuai dengan kebutuhan *user*.

6. Penggunaan Sistem

Tahap terakhir dari metode penelitian ini yaitu penggunaan sistem. Pada tahapan ini sistem yang telah dievaluasi siap untuk digunakan.

2.1.8 *UML (Unified Modeling Language)*

UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan untuk mendokumentasikan, menspesifikasikan, dan mengembangkan perangkat lunak. UML merupakan metode pengembangan sistem berbasis objek dan alat yang mendukung pengembangan sistem (Zulianti et al., 2020).

Menurut definisi lain, UML merupakan alat perancangan sistema yang berorientasi pada objek (*object oriented*) menggunakan pola UML. Sejumlah penelitian telah membuktikan bahwa UML adalah bahasa pemodelan populer yang memiliki visualisasi sistem dan kinerja dokumentasi yang baik. Pemodelan UML bahkan dapat menghasilkan kode-kode pemrograman yang siap diimplementasikan (Nistrina & Sahidah, 2022)

Berdasrkan pengertian di atas, peneliti dapat menyimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan dalam

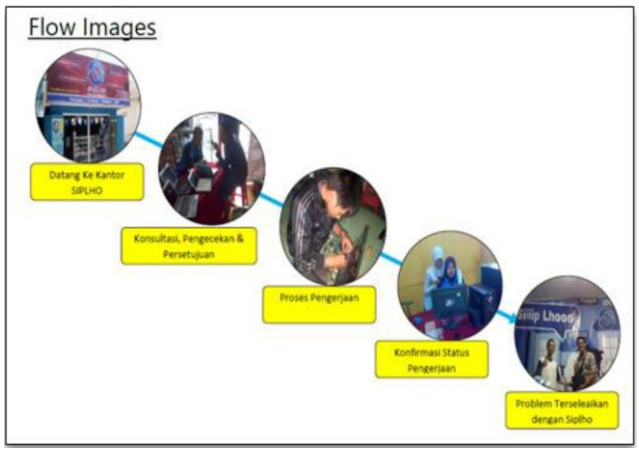
pengembangan sistem berbasis objek untuk mendokumentasikan, merancang, dan menspesifikasikan perangkat lunak. UML membantu memvisualisasikan struktur dan perilaku sistem secara jelas dan sistematis. Sebagai alat perancangan yang populer, UML tidak hanya mempermudah proses dokumentasi, tetapi juga dapat mendukung pembuatan kode program yang siap diimplementasikan.

2.1.8.1 Rich Picture Diagram

Rich Picture merupakan alat pencatatan yang fungsinya mirip dengan narasi tertulis. Media ini menyampaikan informasi melalui gambar, sehingga dapat membantu peneliti memahami konteks dari suatu penelitian. Gambar-gambar yang disusun akan dikaitkan dengan situasi permasalahan yang muncul dalam suatu kasus tertentu. (A. Hidayat et al., 2019).

Dalam pengertian lain, *rich picture* adalah gambar kartun yang menggambarkan sistem yang kompleks secara keseluruhan sehingga mudah dipahami dari berbagai sudut pandang dan memberikan gambaran instan tentang berbagai aspek yang ada (Ningsih et al., 2022).

Dari pengertian di atas, peneliti dapat menyimpulkan bahwa *rich picture* adalah sebuah alat bantu visual yang berfungsi untuk merepresentasikan situasi atau sistem yang kompleks melalui gambar. Alat ini memiliki peran serupa dengan narasi dalam menyampaikan informasi, namun dilakukan secara visual untuk mempermudah pemahaman konteks permasalahan dalam suatu penelitian. Dengan menggabungkan berbagai elemen dan sudut pandang, *Rich Picture* memberikan gambaran menyeluruh dan instan terhadap sistem atau permasalahan yang sedang dikaji.



Gambar 2. 9 Contoh *Rich Picture* (Khoiriyah & Priana, 2020).

2.1.8.2 Flowchart

Flowchart atau diagram alir adalah penggambaran secara grafik dari langkah-langkah dan urutanurutan prosedur dari suatu program. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternative-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan di evaluasi lebih lanjut. Perangkat lunak pembelajaran implementasi *flowchart* ini dapat membantu dalam memahami algoritma-algoritma yang ada dalam menampilkan proses dan tahapan-tahapan dalam pembuatan *flowchart* (Zalukhu et al., 2023)

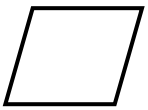
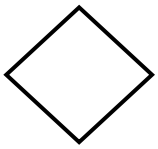
Menurut definisi lain, *flowchart* adalah diagram yang menunjukkan aliran dalam sebuah program serta menggambarkan langkah-langkah untuk menyelesaikan masalah. Diagram ini membantu dalam merancang, menganalisis, dan memahami logika suatu proses dengan lebih terstruktur dan sistematis (Murtianingsih, 2024).

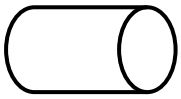
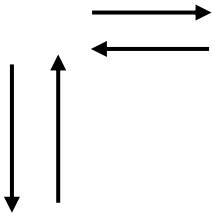
Flowchart atau diagram alir merupakan representasi grafis dari langkah-langkah dan urutan prosedur dalam suatu program atau proses. *Flowchart* mempermudah analisis dan pemecahan masalah dengan membaginya menjadi bagian-bagian yang lebih kecil dan sistematis. Selain itu, *flowchart* juga membantu dalam merancang, memahami, serta

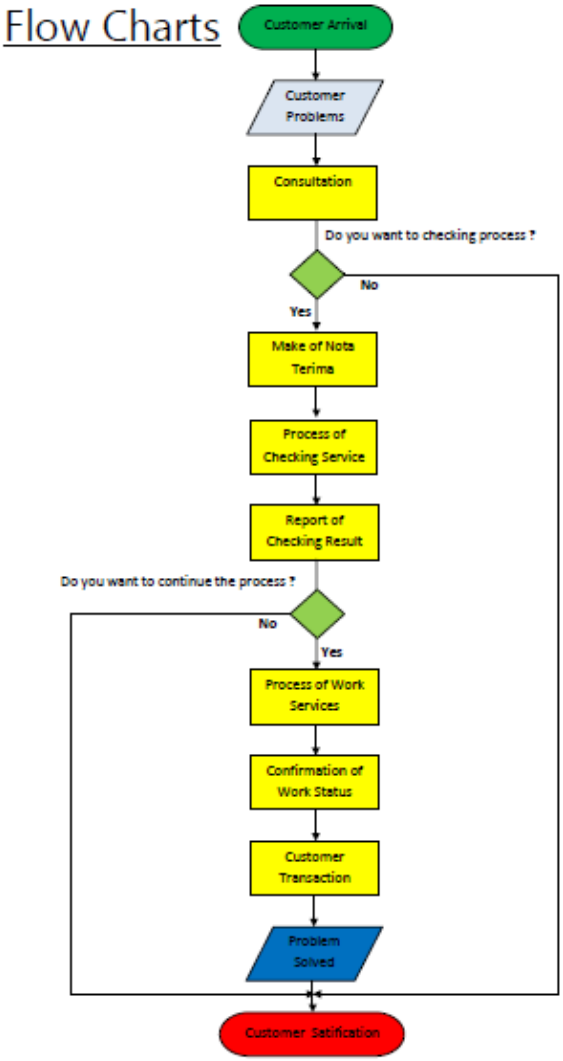
mengevaluasi algoritma atau logika proses secara lebih terstruktur. Penggunaan perangkat lunak pembelajaran untuk implementasi *flowchart* sangat bermanfaat dalam meningkatkan pemahaman terhadap algoritma dan tahapan proses dalam pengembangan program.

Adapun simbol-simbol *flowchart* dapat dilihat pada tabel 2.3 :

Tabel 2. 5 Simbol-Simbol *Flowchart* (Zalukhu et al., 2023)

Simbol	Nama	Fungsi
	*Terminal*	Dipergunakan untuk memulai atau mengakhiri program.
	*Input/Output*	Dipergunakan untuk menyatakan input atau output tanpa melihat jenisnya.
	*Manual Operation*	Digunakan untuk menunjukan pengolahan yang tidak dilakukan oleh komputer.
	*Decision*	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
	*Proccesing*	Digunakan untuk menunjukan pengelolaan data yang dilakukan oleh komputer.

Simbol	Nama	Fungsi
	*Disk Storage*	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari disk.
	*Flow Direction Symbol/Connecting Line*	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.



Gambar 2. 10 Contoh *Flowchart* (Khoiriyah & Priana, 2020).

2.1.8.3 Use Case Diagram

Use case diagram adalah pemodelan untuk menggambarkan perilaku (*behavior*) dari sistem informasi yang akan dikembangkan. *Use case* menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Aditya & Pranatawijaya, 2021).

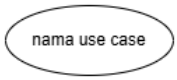
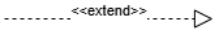
Menurut definisi lain, *use case diagram* adalah metode yang digunakan untuk merepresentasikan interaksi antara aktor dan sistem, atau

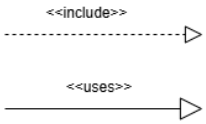
teknik untuk menggambarkan persyaratan fungsional dari suatu sistem (S. Hidayat, 2024).

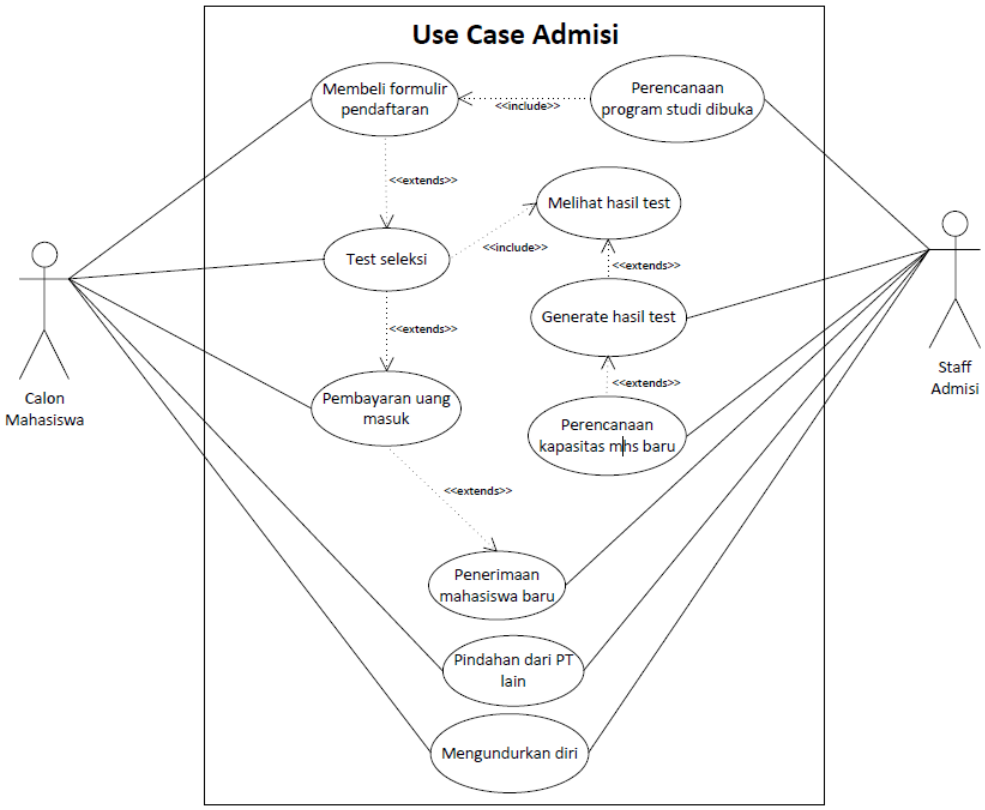
Use case diagram berfungsi untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun, dan dapat menggambarkan fungsi apa saja yang ada pada sebuah sistem informasi (Nistrina & Sahidah, 2022).

Dari pengertian di atas, peneliti dapat menyimpulkan bahwa *Use case diagram* adalah alat pemodelan yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem informasi yang akan dikembangkan. Diagram ini berfungsi untuk merepresentasikan perilaku sistem serta mendeskripsikan kebutuhan atau fungsi-fungsi utama dari sistem tersebut. Dengan demikian, *use case diagram* membantu dalam memahami persyaratan fungsional sistem secara jelas dan terstruktur. Adapun simbol-simbol *use case diagram* dapat dilihat pada tabel 2.4 :

Tabel 2. 6 Simbol-Simbol *Use Case Diagram* (Nistrina & Sahidah, 2022)

Simbol	**Deskripsi**
Usecase 	Fungsionalitas yang ada di sistem
Actor 	Entitas yang berinteraksi dengan sistem
Association 	Interaksi antara aktor dan *use case*
Extend 	Hubungan *use case* tambahan ke *use case* induk. Namun, tanpa *use case*

Simbol	Deskripsi
	tambahan use case induk masih tetap bisa berdiri sendiri.
Generalizattion	Hubungan umum khusus antara dua *usecase*
Menggunakan (*include/use)* 	Hubungan *usecase* tambahan ke suatu *use case* dimana *use case* tersebut bagai syarat untuk menjalankan fungsinya.



Gambar 2. 11 Contoh *Use Case Diagram* (Iskandar, 2021).

2.1.8.4 Skenario Use Case

Skenario *use case* merupakan deskripsi dari masing-masing *use case* yang terdapat dalam diagram *use case*. Setiap skenario ini menggambarkan

urutan tindakan atau langkah-langkah yang dilakukan oleh aktor saat berinteraksi dengan sistem. Umumnya, skenario *use case* disajikan dalam format tabel (Nistrina & Sahidah, 2022)

Menurut definisi lain, skenario *use case* adalah alur kejadian dari *use case* utama yang menjelaskan urutan interaksi antara aktor dan *use case* tersebut, mulai dari awal hingga akhir proses interaksi (Guntara, 2022).

Berdasarkan uraian di atas, skenario *use case* adalah deskripsi rinci dari masing-masing use case dalam diagram use case yang menggambarkan urutan tindakan atau langkah-langkah interaksi antara aktor dan sistem. Skenario ini menjelaskan alur kejadian dari awal hingga akhir proses interaksi dan biasanya disajikan dalam format tabel untuk mempermudah pemahaman alur sistem secara sistematis dan terstruktur.

Tabel 2. 7 Contoh Skenario *Use Case* registrasi

Identifikasi	
No *Use Case*	01
Nama	*Registrasi*
Tujuan	Melakukan registrasi untuk memvalidasi data dan memiliki hak akses ke dalam system
Aktor	*Customer*
Skenario utama	
Kondisi awal	Menampilkan form registrasi
Kondisi akhir	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	Reaksi Sistem

1. Menginputkan *data registrasi*	
	2. Memverifikasi kode *Customer*
	3. Menampilkan halaman utama .
Skenario Alternatif	
	1. Kode *Customer* tidak sesuai..
	2. Kembali ke halaman *registrasi.*

2.1.8.5 Activity Diagram

activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system yang ada pada perangkat lunak. Perbedaan *use case diagram* dan *activity diagram* adalah untuk *use case* menggambarkan bagaimana aktif berjalan atau aktor Ketika menggunakan sistem untuk melakukan aktivitas. Sedangkan *activity diagram* menggambarkan aktifitas sistem bukan apa yang dilakukan oleh aktor (Nistrina & Sahidah, 2022).

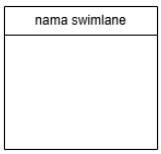
Menurut definisi lain, *activity diagram* adalah representasi dari alur kerja (*workflow*) atau aktivitas dalam suatu sistem yang sedang berjalan, di mana sistem yang dimaksud dalam hal ini adalah *use case* (S. Hidayat, 2024).

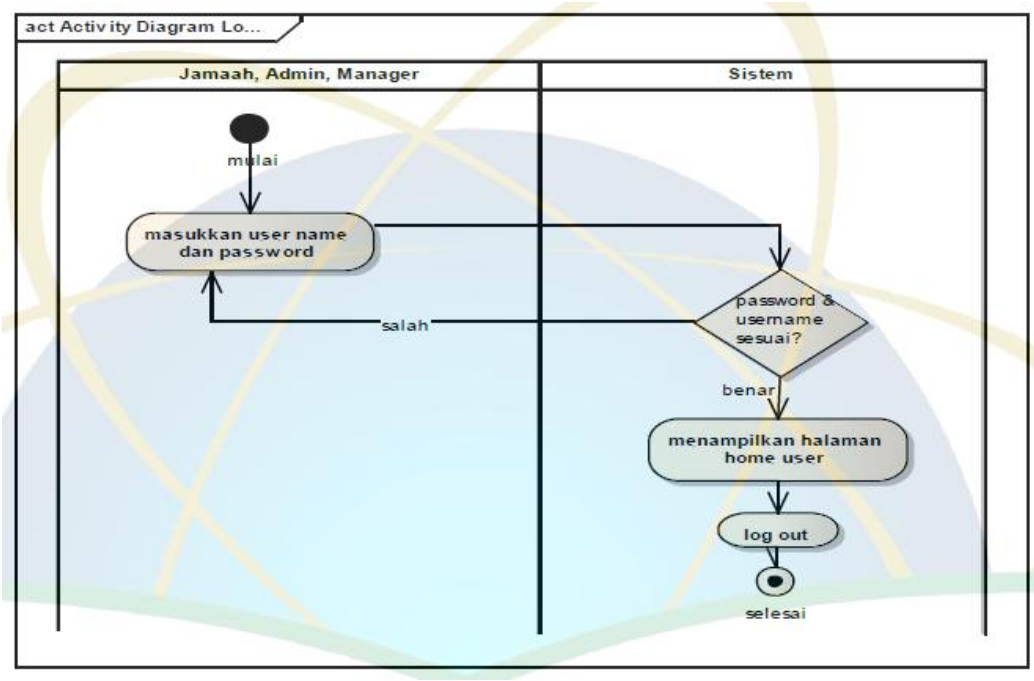
Berdasarkan uraian di atas, dapat peneliti simpulkan bahwa *Activity* diagram adalah representasi visual dari alur kerja atau aktivitas dalam suatu sistem perangkat lunak. Diagram ini menggambarkan proses-proses internal sistem secara rinci, berbeda dengan *use case diagram* yang lebih menekankan pada interaksi antara aktor dan sistem. *Activity diagram* berfungsi untuk memodelkan urutan aktivitas dalam sistem yang sedang

berjalan, sehingga membantu memahami logika proses secara menyeluruh dan terstruktur.

Berikut simbol-simbol dalam *activity diagram,* dapat dilihat pada tabel 2.5 :

Tabel 2. 8 Simbol-Simbol *Activity Diagram* (Nistrina & Sahidah, 2022)

Simbol	**Keterangan**
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memilki satu status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*Decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/*Join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilki satu status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.



Gambar 2. 12 Contoh *Activity Diagram* (Kaisar, 2021).

2.1.8.6 Class Diagram

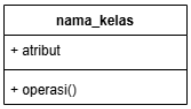
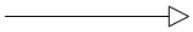
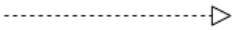
class diagram merupakan pembentukkan utama dari system berorientasi objek yang mempresentasikan suatu class berserta dengan antribut dan operasinya (Nistrina & Sahidah, 2022).

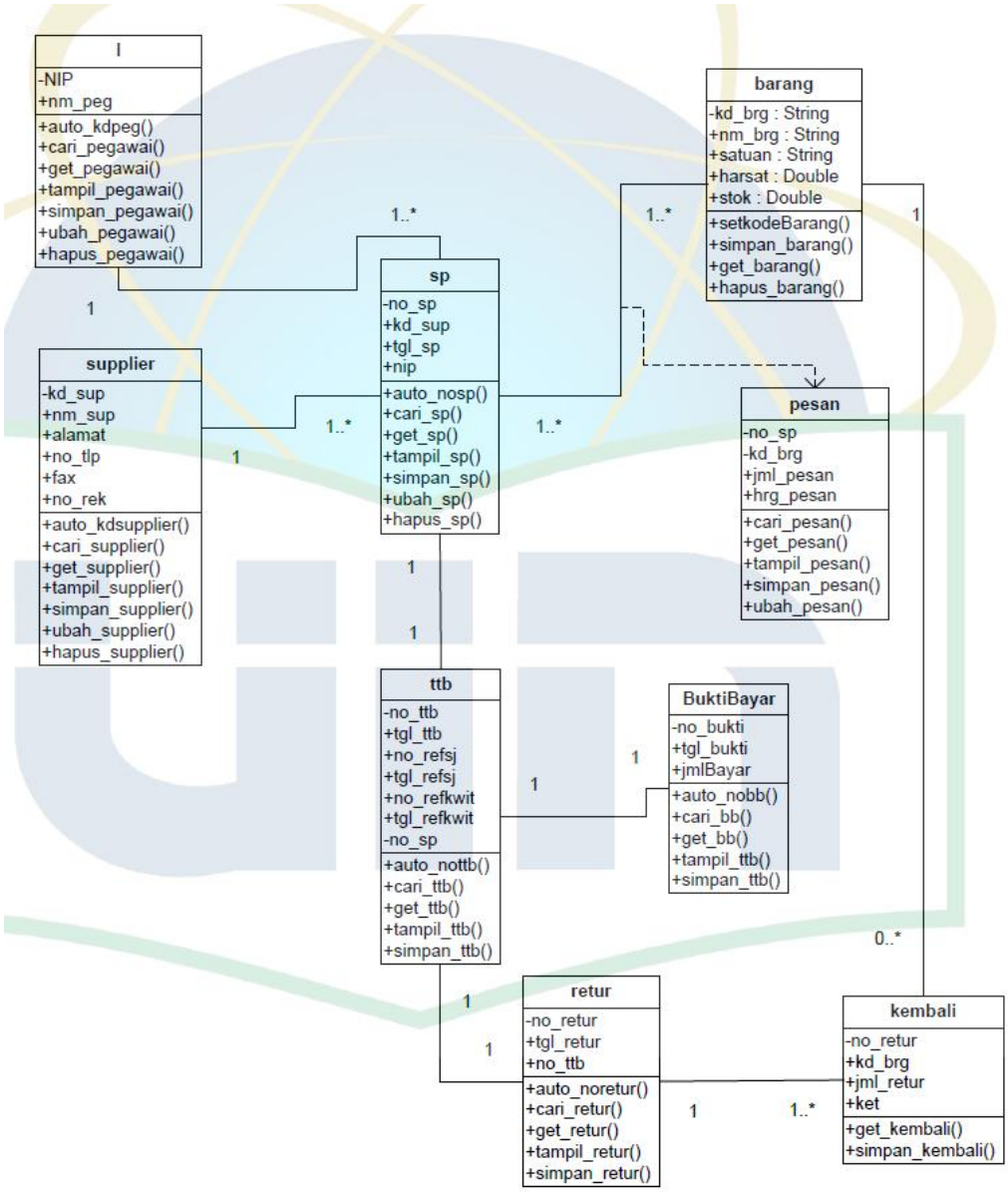
Menurut definisi lain, *class diagram* menggambarkan hubungan antar kelas yang mencerminkan struktur sistem berdasarkan pendefinisian kelas yang akan dibuat dalam pengembangan sistem (S. Hidayat, 2024).

Berdasarkan uraian di atas, dapat peneliti simpulkan bahwa *class diagra*m merupakan pemodelan utama dalam sistem berorientasi objek yang menggambarkan *class*, atribut, dan operasinya. Diagram ini juga menunjukkan hubungan antar *class* yang mencerminkan struktur sistem secara keseluruhan. Dengan demikian, *class diagram* membantu dalam memahami dan merancang arsitektur sistem berdasarkan pendefinisian kelas yang akan digunakan dalam pengembangan perangkat lunak.

. Berikut simbol-simbol yang biasa digunakan dalam *class diagram*, dapat dilihat pada tabel 2.6 :

Tabel 2. 9 Simbol-Simbol *Class Diagram* (Nistrina & Sahidah, 2022)

Simbol	Keterangan
Kelas 	Kelas yang ada distruktur sistem
Antarmuka (*interface)* 	Sama seperti konsep *interface* pada pemograman berorientasi objek.
Asosiasi *(Assosiation)*	Hubungan antar kelas secara umum
Asosiasi berarah (*directed assosiation)*	Hubungan dimana kelas yang satu digunakan pada kelas lain.
Generalisasi 	Hubungan umum – khusus antar kelas.
Kebergantungan *(depedency)* 	Hubungan antar kelas dimana terdapat kebergantungan antar kelas.
Agregasi *(Aggregation)* 	Hubungan antar kelas dengan arti semua bagian *(whole-part)* atau terdapat kelas yang merupakan bagian dari kelas lain.



Gambar 2. 13 Contoh *Class Diagram* (Kaisar, 2021).

2.1.8.7 Sequence Diagram

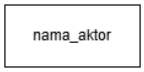
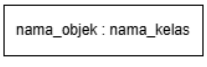
Diagram ini menunjukkan urutan pesan yang dikirim antar objek dalam suatu sistem, sehingga membantu dalam memahami alur komunikasi dan keterkaitan antar komponen secara lebih jelas (S. Hidayat, 2024).

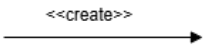
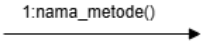
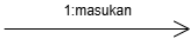
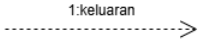
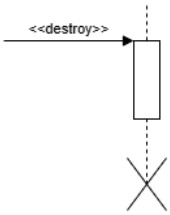
Sequence diagram adalah diagram yangdibuat untuk mengetahui alur dari interaksiantar objek dengan menggambarkan kelakuanobjek pada

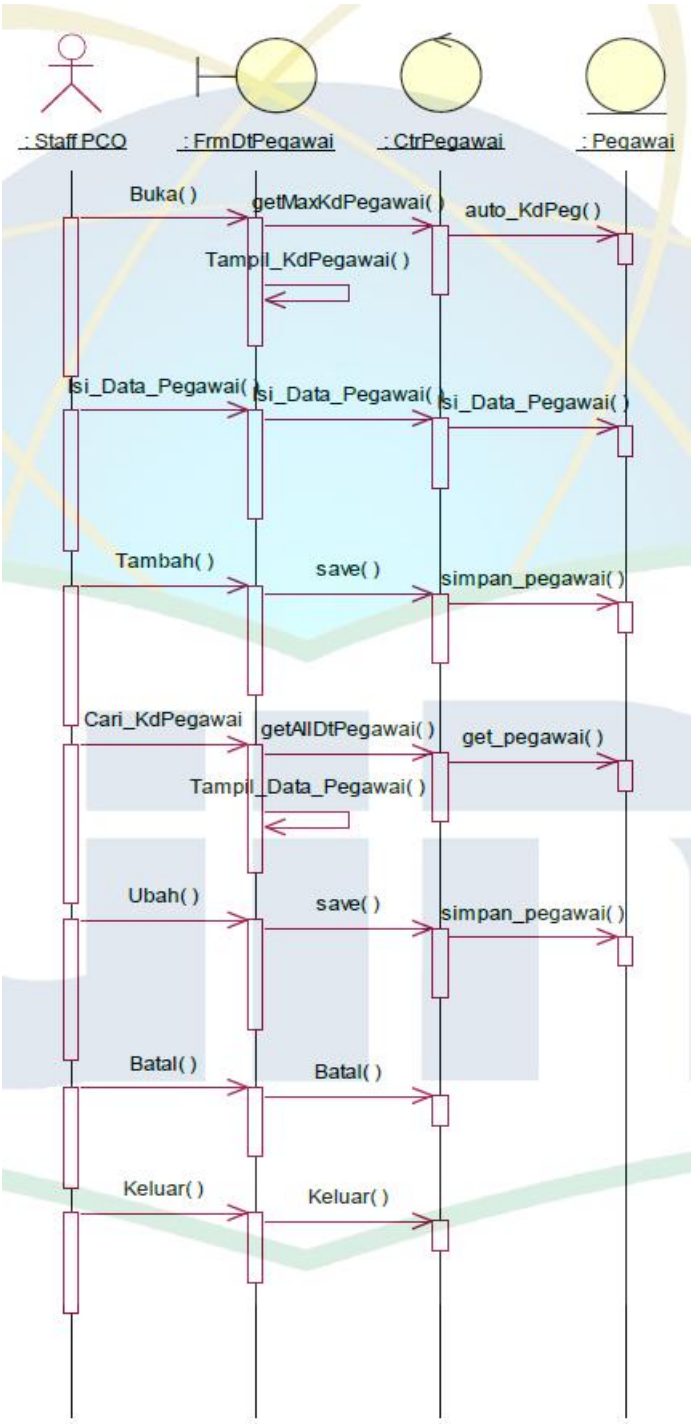
usecase dengan mendeskripsikanwaktu hidup objek dan pesan yang dikirimkan (Tiga et al., 2021).

Berdasarkan dua pengertian di atas, peneliti dapat menyimpulkan bahwa *Sequence diagram* adalah diagram yang menunjukkan urutan pesan yang dikirim antar objek dalam suatu sistem. Diagram ini membantu memahami alur komunikasi, perilaku objek, serta keterkaitan antar komponen berdasarkan use case. Selain itu, sequence diagram juga menggambarkan *lifetime* (waktu hidup) objek dan pesan-pesan yang dikirimkan selama proses interaksi berlangsung.. Berikut simbol-simbol yang biasa digunakan dalam *sequence diagram*, dapat dilihat pada tabel 2.7 :

Tabel 2. 10 Simbol-Simbol *Sequence Diagram* (Tiga et al., 2021)

Simbol	**Keterangan**
Aktor  Atau  Tanpa waktu aktif	Entitas yang berinteraksi dengan sistem
Garis hidup *(lifeline)* 	Menjelaskan kehidupan dari suatu objek
Objek  	Menjelaskan objek yang berinteraksi pesan
Waktu aktif	Menjelaskan objek dalam keadaan aktif dan terlibat dalam interaksi

Simbol	Keterangan
	
Pesan tipe c*reate* 	Menjelaskann objek membuat objek lain dan arah panah menunjukan kepada objek lain tersebut
Pesan tipe *call* 	Menjelaskan objek memanggil sebuah operasi atau metode baik yang ada diobjek itu sendiri atau ada di objek lain.
Pesan tipe *send* 	Menjelaskan obejk mengirim pesan
Pesan tipe *return* 	Menjelaskan objek mengiirm nilai kembalian atau nilai hasil proses
Pesan tipe *destroy* 	Menjelaskan objek mengakhiri hidup objek lain



Gambar 2. 14 Contoh *Sequence Diagram* (Kaisar, 2021).

2.1.9 Android

Android adalah sistem operasi seluler yang berbasis pada versi modifikasi kernel Linux dan perangkat lunak sumber terbuka lainnya, yang dirancang terutama untuk perangkat seluler layar sentuh seperti ponsel pintar dan tablet. Ini adalah sistem operasi untuk perangkat berdaya rendah yang beroperasi dengan baterai dan penuh dengan perangkat keras seperti penerima Sistem Pemosisian Global (GPS), kamera, sensor cahaya dan orientasi, konektivitas Wi-Fi dan LTE (telepon 4G), serta layar sentuh. Seperti semua sistem operasi, Android memungkinkan aplikasi untuk memanfaatkan fitur perangkat keras melalui abstraksi dan menyediakan lingkungan yang ditentukan untuk aplikasi (Goel & Singal, 2021).

Platform ini dibuat oleh Android Inc. yang dibeli oleh Google dan dirilis sebagai Proyek Sumber Terbuka Android (AOSP) pada tahun 2007. Sejak saat itu, Android telah mengalami banyak pembaruan yang secara bertahap meningkatkan sistem operasi, menambahkan fitur baru, dan memperbaiki *bug* pada rilis sebelumnya. Setiap rilis utama diberi nama berdasarkan urutan abjad berdasarkan makanan penutup atau makanan manis, dengan beberapa versi Android pertama disebut "*Cupcake", "Donut", "Eclair", dan "Froyo*", dalam urutan tersebut. Selama pengumuman Android KitKat pada tahun 2013, Google menjelaskan bahwa "Karena perangkat ini membuat hidup kita begitu manis, setiap versi Android dinamai berdasarkan makanan penutup". Ini berubah setelah *Android Pie*(Goel & Singal, 2021).

2.1.10 Bahasa Pemrograman

Bahasa pemrograman komputer adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Tujuan dari pemrograman adalah untuk membuat suatu program yang dapat melakukan suatu perhitungan atau pekerjaan sesuai dengan keinginan yang membuat pemrogram. Pemrograman dapat diartikan sebagai proses menulis, menguji dan memperbaiki (*debug*), dan memelihara kode yang membangun sebuah program dimana kode ini ditulis dalam berbagai bahasa pemrograman. Bahasa pemrograman komputer pertama yaitu bahasa mesin yang terdiri dari bilangan angka-angka, seiring berjalannya waktu

bahasa pemrograman dikembangkan agar mempermuda untuk membuat suatu program (Rochmawati, 2022).

Sejarah bahasa pemrograman beriringan dengan perkembangan mesin hitung dan komputer. Charles Babbage, terinspirasi oleh Wilhelm Schickard, Blaise Pascal, dan Gottfried Leibniz, merancang Mesin *Difference Engine* untuk mengurangi kesalahan perhitungan manual. Algoritma pertama dibuat oleh Ada Lovelace pada tahun 1883 untuk menghitung bilangan Bernoulli dengan mesin tersebut, Pada tahun 1942, ENIAC dikembangkan oleh pemerintah AS, menggantikan gerakan fisik dengan sinyal listrik. John Von Neumann memperkenalkan konsep *shared program technique* dan *conditional control transfer,* yang memungkinkan kode komputer menjadi lebih fleksibel dan efisien (Rochmawati, 2022).

Pada tahun 1949, bahasa *Short Code* muncul sebagai bahasa pemrograman pertama untuk perangkat elektronik, meskipun masih memerlukan konversi manual ke biner. Tahun 1951, Grace Hopper mengembangkan *compiler*, yang mempercepat pemrograman dengan menerjemahkan kode ke dalam format yang dimengerti komputer, Pada tahun 1940-an, komputer bertenaga listrik pertama mulai digunakan, tetapi keterbatasan memori memaksa penggunaan bahasa rakitan. Konrad Zuse mengusulkan Plankalkül (1943-1945) sebagai bahasa tingkat tinggi pertama, tetapi belum diimplementasikan pada masanya (Rochmawati, 2022).

2.1.10.1 Bahasa Pemrograman Java

Bahasa Pemrograman Java dikembangkan pada tahun 1991 oleh lima ahli komputer – Ed Frank, Mike Sheridan, Chris Warth, Patrick Naughton, dan James Gosling yang semuanya bekerja untuk Sun Microsystems Inc. Mereka membutuhkan waktu 18 bulan untuk mengembangkan program, yang awalnya bernama “Oak.” Itu berganti nama menjadi Java pada tahun 1995 karena masalah hak cipta. Konsepnya adalah untuk membuat bahasa pemrograman yang dapat digunakan di seluruh platform dan yang dapat membangun perangkat lunak tertanam untuk elektronik konsumen (Budi Hartono, 2022).

Java menjadi populer seiring dengan berkembangnya *World Wide Web* dan *Internet* karena kemampuannya yang netral terhadap arsitektur, sehingga dapat berjalan di berbagai sistem operasi tanpa bergantung pada perangkat keras atau perangkat lunak tertentu. Java awalnya dikembangkan untuk elektronik konsumen, tetapi kemudian diadopsi secara luas sebagai bahasa pemrograman untuk web karena sifatnya yang ringkas dan mudah dibawa melalui jaringan. Kini, Java telah berkembang menjadi teknologi yang kuat, aman, *multi-utas*, berorientasi objek, serta mendukung distribusi dan portabilitas (Budi Hartono, 2022).

Java memiliki beberapa fitur utama yang menjadikannya bahasa pemrograman yang kuat dan populer. Java sederhana karena tidak memiliki fitur kompleks seperti *pointer* dan pewarisan ganda. Keamanannya tinggi dengan *firewall virtual* dan pembatasan akses melalui *Java Runtime Environment* (JRE). Java juga portabel dan platform-independen, memungkinkan kode berjalan di berbagai sistem operasi. Java mendukung *multi-threading*, memungkinkan eksekusi beberapa tugas secara bersamaan. Sebagai bahasa berorientasi objek, Java mengadopsi konsep OOP seperti *polimorfisme,* pewarisan, enkapsulasi, dan abstraksi. Selain itu, Java dapat digunakan untuk mengembangkan aplikasi terdistribusi melalui *Enterprise Java Beans* (EJB) dan *Remote Method Invocation* (RMI). Java juga kokoh dengan sistem penanganan pengecualian dan pengumpulan sampah untuk mengelola kesalahan runtime serta memori secara otomatis (Budi Hartono, 2022).

2.1.10.2 Bahasa Pemrograman Kotlin

Kotlin adalah bahasa pemrograman open source berjenis statis yang mendukung pemrograman berorientasi objek dan fungsional. Kotlin memberikan sintaksis dan konsep serupa dari bahasa lain, di antaranya termasuk C#, Java, dan Scala. Kotlin tidak dimaksudkan agar unik, melainkan mengambil inspirasi dari perkembangan bahasa selama puluhan tahun. Kotlin ada dalam varian yang menargetkan JVM (Kotlin/JVM),

JavaScript (Kotlin/JS), dan kode native (Kotlin/Native)(Rosida Vivin, Nahari Riza, 2024).

Kotlin dikelola oleh Kotlin Foundation, sebuah grup yang didirikan oleh JetBrains dan Google, yang ditugaskan untuk mengembangkan dan melanjutkan pengembangan bahasa. Kotlin secara resmi didukung oleh Google untuk pengembangan Android. Artinya, dokumentasi dan alat Android didesain dengan mempertimbangkan Kotlin, Android API tertentu, seperti Android KTX, khusus untuk Kotlin, tetapi sebagian besar ditulis dalam Java dan dapat dipanggil dari Java atau Kotlin. Interoperabilitas Kotlin dengan Java adalah inti dari pertumbuhannya (Rosida Vivin, Nahari Riza, 2024).

Kesimpulannya, Kotlin adalah bahasa pemrograman open source berjenis statis yang menggabungkan paradigma berorientasi objek dan fungsional, dengan sintaksis yang terinspirasi dari bahasa populer seperti Java, C#, dan Scala. Bahasa ini dikembangkan oleh *Kotlin Foundation* (JetBrains dan Google) dan mendapatkan dukungan resmi dari Google untuk pengembangan Android, sehingga membuatnya sangat relevan bagi ekosistem Android. Dengan dukungan interoperabilitas penuh terhadap Java, Kotlin semakin berkembang sebagai bahasa utama dalam pengembangan aplikasi modern.

2.1.10.3 Bahasa Pemrograman PHP

Menurut Solichin, PHP merupakan salah satu bahasa pemograman berbasis web yang ditulis oleh dan untuk pengembang web. PHP pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *softwere* dan anggota tim Apache, dan dirilis pada akhir tahun 1994. PHP dikembangkan dengan tujuan awalmhanya untuk mencatat pengunjung pada web sitem pribadi Rasmus Lerdorf. PHP merupakan bahasa pemorgaman berbasis web yang dibuat secara khusus untuk membangun aplikasi berbasis web. Selain tersedia secara gratis, PHP juga mudah dipelajari oleh siapapun (A. Hidayat et al., 2019).

PHP merupakan bahasa *server-side* yang menyatu dengan html, untuk membuat halaman web yang dinamis, salah satu fungsinya adalah untuk menerima dan mengolah dan menampilkan data ke sebuah situs, data yang diterima akan diolah disebuah prorgam *database server*, untuk kemudian hasilnya ditampilkan kembali ke *layer browser* sebuah situs (A. Hidayat et al., 2019).

PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di *internet*. Sedangkan dalam pengertian lain PHP adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web serverside* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server* (A. Hidayat et al., 2019).

Menurut Tim EMS, PHP adalah singkatan dari *Hypertext Processor.* Saat pertama kali dikembangkan oleh *programmer* bernama Rasmus Lerdoft, PHP awalnya adalah singkatan dari *Personal Home Page Tools*. Namun setelah dikembangkan oleh Zeev Suraski dan Andi Gutmans. Dan fiturnya bertambah, maka PHP diubah singkatannya menjadi sekarang ini. PHP merupakan jenis bahasa *scripting* yang lazim digunakan di halaman web. Artinya, kode ini langsung dimasukan kedalam kode HTML (A. Hidayat et al., 2019)

Menurut Tim EMS, Salah satu kelebihan PHP adalah kemudahannya untuk berintegrasi dengan *database*. PHP dapat mendukung beberapa *database* secara langsung tanpa menginstal konektor seperti halnya bahasa pemrograman Java. Dengan demikian PHP sangat Fleksibel berhubungan dengan berbagai database. Dari beberapa database, paling banyak disandingakan dengan PHP adalah MySQL. Untuk menghubungkan PHP dengan *database*, hanya perlu mengetahui nama *database* dan lokasinya, serta *username* dan *password* untuk menuju ke *database* tersebut (A. Hidayat et al., 2019).

2.1.11 MySQL

MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang populer, terutama digunakan dalam pemrograman web. Menggunakan SQL sebagai bahasa dasar, MySQL tersedia di berbagai *platform* seperti Linux dan Windows. Sistem ini memiliki dua jenis lisensi, yaitu lisensi gratis berbasis *open source* (GPL) yang memungkinkan pengguna memodifikasi dan membagikan kode sumbernya, serta lisensi komersial yang memerlukan pembayaran untuk layanan tambahan. Setelah diakuisisi oleh *Oracle* pada tahun 2009, sempat muncul kekhawatiran bahwa MySQL akan dihentikan, namun hingga kini tetap tersedia dan digunakan secara luas (Kurniati et al., 2020).

MySQL dapat diakses melalui dua tampilan, yaitu berbasis GUI yang mempermudah penggunaan tanpa banyak perintah, serta berbasis console yang lebih cocok bagi pemula untuk memahami SQL secara mendalam. MySQL memiliki berbagai kelebihan, seperti kecepatan tinggi dalam pengolahan data, kemudahan penggunaan, serta sifat *open source* yang memungkinkan pengembangan lebih lanjut oleh komunitas. Selain itu, MySQL mampu menangani jumlah data yang besar, memiliki sistem keamanan yang baik, mendukung berbagai sistem operasi, serta dapat digunakan tanpa biaya yang tinggi. Namun, MySQL juga memiliki beberapa kekurangan, seperti kurangnya dukungan untuk aplikasi *visual* atau *desktop*, pembaruan yang lebih lambat dibandingkan dengan RDBMS berbayar seperti Oracle atau MS SQL Server, serta kemampuan pengolahan data berskala besar yang masih menjadi perdebatan (Kurniati et al., 2020).

2.1.12 Tools Perangkat Lunak

2.1.12.1 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan Android dari Google yang dikembangkan berdasarkan *IntelliJ.* Dibuat di atas *IntelliJ* IDEA karya JetBrains, Android Studio dirancang khusus untuk pengembangan aplikasi Android. IDE ini

dapat digunakan di sistem operasi Windows, Mac OS X, dan Linux (Erni Sri Wahyuni, 2020)

Menurut definisi lain, android studio adalah sebuah lingkungan pengembangan perangkat lunak terpadu (*Integrated Development Environment/IDE*) yang dirancang khusus untuk membangun aplikasi Android, dan dikembangkan dengan basis IntelliJ IDEA. Selain berfungsi sebagai editor kode yang canggih dan menyediakan berbagai alat bantu pengembangan dari *IntelliJ*, android studio juga menyuguhkan beragam fitur tambahan yang dirancang untuk mendukung peningkatan produktivitas selama proses pembuatan aplikasi Android (Fadhilla et al., 2021).

Berdasarkan definisi diatas, peneliti dapat menyimpulkan Android Studio adalah *Integrated Development Environment* (IDE) yang dirancang khusus untuk pengembangan aplikasi Android, berbasis pada IntelliJ IDEA. IDE ini dirancang khusus untuk meningkatkan produktivitas dalam pembuatan aplikasi Android serta dapat digunakan di berbagai sistem operasi seperti Windows, macOS, dan Linux.

2.1.12.2 *Visual Studio Code*

Visual Studio Code adalah editor teks ringan dan andal yang dikembangkan oleh Microsoft untuk sistem operasi *multiplatform*, yang berarti tersedia untuk Linux, Mac, dan Windows. Editor ini mendukung bahasa pemrograman seperti *Javascript, Typescript*, dan *Node.js* secara langsung, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat diunduh melalui marketplace Visual Studio Code, seperti C++, C#, Python, Go, Java, PHP, dan lainnya (Ningsih et al., 2022).

Sedangkan menurut definisi lain, *Visual Studio Code* merupakan editor kode sumber yang dikembangkan oleh Microsoft dan dapat digunakan pada Windows, Linux, serta macOS. Editor ini mendukung berbagai fitur, seperti *debugging*, integrasi dengan Git dan GitHub, penyorotan sintaksis, penyelesaian kode otomatis, snippet, serta refactoring kode. Selain itu, Visual Studio Code memiliki fleksibilitas tinggi, memungkinkan pengguna

untuk menyesuaikan tema, pintasan *keyboard*, preferensi, serta menambahkan fungsi tambahan melalui pemasangan ekstensi (Nurmufid et al., 2023).

Berdasarkan definisi di atas, peneliti dapat menyimpulkan *Visual Studio Code* adalah editor kode sumber dari Microsoft yang mendukung berbagai bahasa pemrograman dan fitur seperti *debugging*, integrasi Git, serta ekstensi untuk menambah fungsionalitas.

2.1.12.3 XAMPP

XAMPP adalah paket perangkat lunak yang mencakup Apache HTTP Server, MySQL, dan PhpMyAdmin, yang digunakan untuk membangun dan mengelola server lokal dalam pengembangan aplikasi berbasis web (Zulianti et al., 2020).

Menurut pengertian lain, XAMPP merupakan perangkat lunak yang mudah digunakan, gratis, dan mendukung instalasi di sistem operasi Linux dan Windows. Keuntungan lainnya adalah dengan hanya menginstal satu kali, pengguna sudah dapat mengakses Apache Web Server, MySQL Database Server, dukungan PHP (PHP 4 dan PHP 5), serta beberapa modul tambahan lainnya (Ningsih et al., 2022).

Berdasarkan definisi di atas, peneliti dapat menarik kesimpulan XAMPP adalah paket perangkat lunak gratis yang mendukung berbagai sistem operasi dan berisi Apache, MySQL, serta PhpMyAdmin. Perangkat ini digunakan untuk membangun dan mengelola *server* lokal dalam pengembangan aplikasi *web* secara mudah dan praktis.

2.1.12.4 *IBM Rational Rose*

Rational Rose merupakan alat pemodelan visual yang berfungsi dalam analisis dan perancangan sistem perangkat lunak berorientasi objek. Alat ini digunakan untuk memodelkan sistem sebelum proses pengkodean dimulai, sehingga arsitektur sistem dapat dipastikan sudah dirancang dengan baik sejak awal (Beno et al., 2022).

Menurut definisi lainn, *Rational Rose* merupakan alat pemodelan visual yang digunakan untuk mendukung analisis dan perancangan sistem perangkat lunak berbasis objek. Dengan *Rational Rose*, sistem dapat dimodelkan sebelum proses *coding* dilakukan, sehingga memastikan bahwa arsitektur sistem sudah terstruktur dengan baik sejak awal (Riskawati, et al., 2021).

Berdasarkan definisi di atas, peneliti dapat menyimpulkan bahwa *Rational Rose* adalah alat pemodelan visual yang digunakan untuk analisis dan perancangan sistem perangkat lunak berbasis objek. Dengan alat ini, sistem dapat dimodelkan sebelum proses pengkodean dimulai, sehingga memastikan arsitektur sistem telah dirancang dengan baik sejak awal.

Contoh penggunaan *Rational Rose* dalam penelitian ini adalah untuk membuat diagram *Unified Modeling Language* (UML), seperti *use case diagram, activity diagram, class diagram* dan *sequence diagram*, yang digunakan untuk menggambarkan kebutuhan fungsional sistem serta alur interaksi antara aktor dan sistem dalam pengembangan aplikasi. Dengan *Rational Rose*, peneliti dapat memvisualisasikan rancangan sistem secara lebih jelas dan terstruktur sebelum proses implementasi dilakukan.

2.1.13 Tools Pengujian

2.1.13.1 *Black Box Testing*

Pengujian *black box* merupakan salah satu metode pengujian sistem yang berfokus pada sisi fungsionalitas atau kegunaan dari sistem secara keseluruhan. Dalam pengujian ini, penguji tidak perlu mengetahui struktur internal atau kode program, melainkan hanya mengevaluasi apakah *input* yang diberikan menghasilkan *output* yang sesuai dengan yang diharapkan. Dengan kata lain, pengujian ini dilakukan untuk memastikan bahwa sistem mampu merespon berbagai masukan dengan hasil keluaran yang benar sesuai spesifikasi yang telah ditentukan (Fazrin et al., 2021).

Sementara itu, menurut pengertian lain, pengujian *Black box* adalah metode pengujian yang digunakan untuk menguji fungsi-fungsi dari

perangkat lunak yang telah dirancang tanpa memperhatikan struktur internal atau kode programnya. Pengujian ini berfokus pada keluaran yang dihasilkan berdasarkan *input* yang diberikan, sehingga memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang ditentukan. (Nahar Mardiyantoro & Listiani, 2021).

Berdasarkan definisi diatas, peneliti dapat menyimpulkan bahwa Pengujian *Black Box* adalah metode pengujian perangkat lunak yang mengutamakan pengujian fungsi dan kesesuaian program tanpa memerlukan pengetahuan tentang struktur internal program. Berikut ini contoh pengujian *Blackbox* :

Tabel 2. 11 Contoh Pengujian *Blackbox* (Nahar Mardiyantoro & Listiani, 2021)

No	**Pengujian**	**Aktivitas Pengujian**	**Hasil Pengujian**	**Kesimpulan**
1.	Pembuatan *QR-Code*	Cetak KTA menggunakan aplikasi, Scan *QR-Code* dengan *Barcode Scanner*	Muncul *link* pengecekan KTA	OK
2.	Enkripsi Algoritma AES	Cek *enkripsi link* yang diperoleh dari *Barcode* dengan hasil *enkripsi* secara manual. Hasil *enkripsi* dari perhitungan manual :	*link* diperoleh : http://192.168.1.9/kta/kta_detail.php?id=**b87440a869d082040177b1af3c7158c6** *Enkripsi* menggunakan aplikasi maupun	OK

		b87440a869d08204 077b1a f3c7158c6	perhitungan secara manual menghasilkan *chipertext* yang sama	
3.	Dekripsi Algoritma AES	Klik *link* yang muncul setelah memindai *QR-Code*	*Link* mengarahkan ke halaman pengecekan KTA. Data yang ditampilkan sama dengan data KTA yang dipindai *QR-Code* nya	OK

2.1.13.2 White Box Testing

White Box Testing adalah metode pengujian yang memeriksa struktur internal perangkat lunak, termasuk desain dan kode program. Pengujian ini dilakukan dengan menganalisis alur logika, kondisi, loop, dan jalur data dalam sistem untuk memastikan bahwa setiap bagian kode berfungsi dengan benar sesuai dengan rancangan yang telah ditetapkan (Sie et al., 2022).

Sementara itu, pengujian *white box* juga dapat diartikan sebagai metode pengujian yang menggunakan struktur kontrol sebagai bagian dari desain tingkat komponen untuk menyusun test case (Londjo, 2021).

Berdasarkan dua pengertian diatas, peneliti dapat menyimpulkan bahwa *white box testing* adalah metode pengujian perangkat lunak yang fokus pada pengujian struktur internal, rancangan, dan kode program dengan memanfaatkan pemahaman terhadap struktur kontrol aplikasi untuk menyusun *test case*. *White Box* sendiri mempunyai beberapa teknik di dalam

pengujiannya, yaitu *Data Flow Testing, Control Flow Testing*, *Basic Path / Path Testing*, dan *Loop Testing.*

Teknik *Basis Path* adalah metode pengujian *white box* yang dikembangkan oleh Tom McCabe. Metode ini memungkinkan penguji untuk menilai kompleksitas logika dalam desain (Londjo, 2021). Teknik ini melibatkan beberapa elemen, seperti notasi *flow graph* yang digunakan untuk menggambarkan alur kontrol program, *cyclomatic complexity* yang digunakan untuk menghitung jumlah jalur pengujian, *independent path* yang menentukan jalur pengujian yang harus dijalani setidaknya satu kali, *graph matrix* berupa matriks dua dimensi untuk mempermudah perhitungan *cyclomatic complexity*, dan *test case* yang disusun berdasarkan jalur *independent path* yang telah ditentukan (Sie et al., 2022).

Untuk menghitung *cyclomatic complexity* digunakan rumus sebagai berikut :

$V(G) = E - N + 2$

E = jumlah garis atau hubungan (*edges*) dalam flowgraph

N = jumlah titik atau simpul (*nodes*) dalam flowgraph

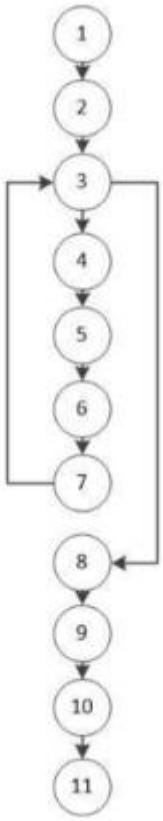
Berikut ini contoh pengujian *white box* aplikasi pengecekan keaslian produk obat herbal PT.Cipta Herba Indonesia (Murtianingsih, 2024) :

1. Pengujian Enkripsi

Tabel 2. 12 Flow Graph (Sie et al., 2022)

Node	*Source Code*
1	public function encrypt($x) { $y = '';
2	$this->addRoundKey(0);
3	for ($i = 1; $i < $this->Nr; $i++) {
4	$this->subBytes();
5	$this->shiftRows();
6	$this->mixColumns();
7	$this->addRoundKey($i); }
8	$this->subBytes();
9	$this->shiftRows();
10	$this->addRoundKey($i);
11	return $y; }

Berdasarkan kode program enkripsi AES-128 tersebut, maka diagram alurnya dapat dilihat pada Gambar 2.10 :



Gambar 2. 15 Flowgraph Diagram Enkripsi

Gambar 2. 16 Flowgraph Diagram Enkripsi AES-128

Dari Gambar 2.8, dapat dihitung *cyclomatic complexity* sebagai berikut :

$$V(G) = (E - N) + 2$$

Keterangan :

V (G) = Cyclomatic Complexity

E = Total jumlah edge

N = Total jumlah node

Maka, dari gambar diketahui bahwa E=11, N=11

V (G) = (11 – 11) + 2

V (G) = 2

Hasil dari perhitungan *cyclomatic complexity* menujukkan jumlah dari *independent path* dari basis *path* atau menunjukkan jumlah pengujian yang baru dijalankan untuk memastikan semua *statement* dalam program dijalankan

minimal sekali. Hasil dari *independent path* nya adalah sebagai berikut :

Path 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 3 – 8 – 9 – 10 – 11

Path 2 : 1 – 2 – 3 – 8 – 9 – 10 – 11

2.1.13.3 UAT *(User Accepttance Testing)*

User Acceptance Testing (UAT) adalah tahapan yang bertujuan untuk memastikan bahwa solusi yang dikembangkan dalam sistem memenuhi kebutuhan pengguna. Berbeda dengan pengujian sistem yang fokus pada memastikan perangkat lunak berjalan lancar tanpa gangguan dan sesuai dengan spesifikasi yang diminta, UAT lebih mengutamakan agar solusi yang ada dapat berfungsi dengan baik bagi pengguna sehingga mereka dapat menerima dan menggunakannya (Suprapto, 2021).

Sementara itu, menurut pengertian lain, UAT adalah proses untuk memverifikasi apakah solusi yang ada dalam sistem sesuai dengan kebutuhan pengguna. Pengujian ini dilakukan pada tahap akhir sebelum sistem resmi diterapkan, dengan melibatkan pengguna akhir untuk memastikan bahwa semua fitur berfungsi sebagaimana mestinya dan memenuhi ekspektasi bisnis. (Bagus Rubyanto Esha Andrai, 2023).

Sehingga dapat disimpulkan dari kedua pernyataan diatas bahwa *User Acceptance Testing (UAT)* adalah tahap akhir pengujian yang bertujuan untuk memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan dan ekspektasi pengguna. Pengujian ini melibatkan pengguna akhir untuk memverifikasi bahwa fitur-fitur dalam sistem berfungsi dengan baik dan dapat diterima sebelum resmi diterapkan.

Pengujian UAT dilakukan dengan mengumpulkan data melalui penggunaan kuesioner. Metode ini memungkinkan pengumpulan informasi yang lebih menyeluruh karena responden dapat memberikan jawaban secara bebas tanpa merasa takut memberikan jawaban yang tidak sesuai dengan

harapan. Kuesioner digunakan untuk mengukur pendapat atau persepsi seseorang (Murtianingsih, 2024).

Pengujian UAT dapat dilakukan dengan menyajikan pertanyaan dalam bentuk kuesioner yang mencakup aspek sistem, pengguna, dan interaksi. Persentase skor aktual dapat dihitung menggunakan rumus berikut (Murtianingsih, 2024) :

$$Persetase\ Kelayakan\ (\%) = \frac{Skor\ hasil\ pengujian}{skor\ yang\ diharapkan} \text{X } 100\%$$

Untuk menilai persentase kelayakann skor dengan melihat acuan pada Tabel 2.9 :

Tabel 2. 13 Persentase Skor Kelayakan

% Jumlah Skor	**Kriteria**
20,00% - 36,00%	Tidak Baik
36,01% - 52,00%	Kurang Baik
52,01% - 68,00%	Cukup
68,01% - 84,00%	Baik
84,01% - 100%	Sangat Baik

Berikut ini contoh kuesioner (Murtianingsih, 2024) :

Tabel 2. 14 Contoh Kuesioner

No	Pertanyaan	Jawaban				
		A	B	C	D	E
1	Apakah dengan adanya aplikasi ini dapat membantu mengetahui informasi mengenai produk?					
2	Apakah menu bantuan penggunaan ini mudah dipahami?					

No	Pertanyaan	Jawaban				
		A	B	C	D	E
3	Apakah tampilan aplikasi ini menarik?					
4	Apakah menu *searcing* dalam *QR-Code* ini mudah dipahami?					
5	Apakah proses *scanning QR-Code* ini sudah lengkap?					
6	Apakah informasi produk dalam *QR-Code* ini sudah lengkap?					
7	Apakah *QR-Code* ini menarik?					
8	Apakah aplikasi ini sudah cukup baik?					

Dari kuesioner tersebut, bobot jawaban dapat dilihat pada Tabel 2.11 :

Tabel 2. 15 Bobot Jawaban

Kode	**Keterangan Jawaban**	**Bobot**
A	Sangat Setuju	5
B	Setuju	4
C	Ragu-ragu	3
D	Cukup Setuju	2
E	Sangat Tidak Setuju	1

Kemudian kuesioner disebarkan ke 30 responden (mahasiswa). Jawaban dan hasil kuesioner, terdiri dari berbagai tingkatan yang dapat dilihat pada tabel 2.12 :

Tabel 2. 16 Contoh Hasil Jawaban Kuesioner

No	Pertanyaan	Jawaban					Persentase				
		A	B	C	D	E	A	B	C	D	E
1	Apakah dengan adanya aplikasi ini dapat membantu mengetahui informasi mengenai produk?	13	8	5	4	0	43%	27%	17%	13%	0%
2	Apakah menu bantuan penggunaan ini mudah dipahami?	14	10	3	3	0	47%	33%	10%	10%	0%
3	Apakah tampilan aplikasi ini menarik?	8	11	6	2	3	27%	37%	20%	7%	10%
4	Apakah menu *searcing* dalam *QR-Code* ini mudah dipahami?	17	9	1	3	0	57%	30%	3%	10%	0%
5	Apakah proses *scanning QR-Code* ini sudah lengkap?	19	8	3	0	0	63%	27%	10%	0%	0%
6	Apakah informasi produk dalam *QR-Code* ini sudah lengkap?	20	7	1	2	0	67%	23%	3%	7%	0%
7	Apakah *QR-Code* ini menarik?	16	8	4	1	1	53%	27%	13%	3%	3%

No	Pertanyaan	Jawaban					Persentase				
		A	B	C	D	E	A	B	C	D	E
8	Apakah aplikasi ini sudah cukup baik?	18	7	3	2	0	60%	23%	10%	7%	0%

Data yang ada pada tabel 2.12 kemudian diolah dengan cara mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan pada tabel 2.11. hasil dari perhitungannya dapat dilihat pada tabel 2.13 :

Tabel 2. 17 Contoh Perhitungan Kuesioner

No	Pertanyaan	Jawaban					Hasil
		Ax5	Bx4	Cx3	Dx2	Ex1	
1	Apakah dengan adanya aplikasi ini dapat membantu mengetahui informasi mengenai produk?	13	8	5	4	0	120
2	Apakah menu bantuan penggunaan ini mudah dipahami?	14	10	3	3	0	125
3	Apakah tampilan aplikasi ini menarik?	8	11	6	2	3	109
4	Apakah menu *searcing* dalam *QR-Code* ini mudah dipahami?	17	9	1	3	0	130

No	Pertanyaan	Jawaban					Hasil
		Ax5	Bx4	Cx3	Dx2	Ex1	
5	Apakah proses *scanning QR-Code* ini sudah lengkap?	19	8	3	0	0	136
6	Apakah informasi produk dalam *QR-Code* ini sudah lengkap?	20	7	1	2	0	135
7	Apakah *QR-Code* ini menarik?	16	8	4	1	1	127
8	Apakah aplikasi ini sudah cukup baik?	18	7	3	2	0	131
Total							1013

Data tersebut kemudian di analisis, hasil analisis dapat dilihat pada Tabel 2.14 :

Tabel 2. 18 Contoh Analisis Kuesioner

No	Pertanyaan	(Jumlah/30)/5	Persentase (x 100%)
1	Apakah dengan adanya aplikasi ini dapat membantu mengetahui informasi mengenai produk?	0.80	80%
2	Apakah menu bantuan penggunaan ini mudah dipahami?	0.83	83%
3	Apakah tampilan aplikasi ini menarik?	0.73	73%

No	Pertanyaan	(Jumlah/30)/5	Persentase (x 100%)
4	Apakah menu *searcing* dalam *QR-Code* ini mudah dipahami?	0.87	87%
5	Apakah proses *scanning QR-Code* ini sudah lengkap?	0.91	91%
6	Apakah informasi produk dalam *QR-Code* ini sudah lengkap?	0.90	90%
7	Apakah *QR-Code* ini menarik?	0.85	85%
8	Apakah aplikasi ini sudah cukup baik?	0.87	87%

Dari Tabel 2.12 diketahui bahwa skor hasil pengujian adalah sebesar 1013. Setelah diketahui skor hasil pengujian, kemudian untuk mengetahui persentase kelayakan adalah sebagai berikut:

$$Persetase\ Kelayakan\ (\%) = \frac{Skor\ hasil\ pengujian}{skor\ yang\ diharapkan} \text{X } 100\%$$

$$= \frac{1013}{1200} x\ 100\%$$

$$= 85\%$$

Keterangan :

- **Skor hasil pengujian** merupakan Total Keseluruhan Hasil Perhitungan Kuisioner dapat dilihat da tabel 2.12.

  Untuk menghitung total keseluruhan hasil perhitungan kuisioner yaitu Hasil Jawaban Kuisioner (dapat dilihat pada tabel 2.11) dikalikan dengan Bobot Jawaban yang telah ditentukan (dapat dilihat pada tabel 2.10). kemudian hasil dari

setiap bobot jawaban ditambah sampai pertanyaan delapan (8), lalu totalkan sehingga hasilnnya menjadi **1013**.

- **Skor yang diharapkan** merupakan total hasil perhitungan kuisioner, dimana dari 30 responden diharapkan menjawab 8 pertanyaan dengan jawaban sangat setuju (bobot jawaban 5). Bisa kita hitung :

  30 (responden) x 5 (bobot jawaban sangat setuju) = 150.

  kemudian 150 (hasil setiap pertanyaan) x 8 (total pertanyaan) = **1.200**.

Kesimpulan :

Berdasarkan hasil perhitungan, aplikasi dinilai layak dengan persentase kelayakan sebesar 85%, yang menunjukkan bahwa mayoritas responden merasa aplikasi ini sudah memenuhi harapan dari segi fungsi, tampilan, dan kemudahan penggunaan.

2.2 Penelitian Sebelumnya *(Previous Work)*

Penelitian-penelitian terdahulu yang relevan dengan penelitian yang dilakukan oleh penyusun saat ini. Terhadap beberapa penelitian tersebut diharapkan dapat menemukan alternatif solusi dengan menggunakan, memodifikasi dan atau menemukan model baru.

Penelitian sebelumnya ini menjadi pembanding atau perbedaan antara penelitian sebelumnya yang pernah ada dan yang akan peneliti lakukan sebagai solusi untuk melakukan penelitian.

Tabel 2. 19 Perbandingan penelitian sebelumnya

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
1.	Rancangan Prototipe Marker Beacon sebagai Media Pembelajaran di Politeknik Penerbangan Medan (Christine et al., 2024).	Taruna/I di Politeknik Penerbangan Medan mengalami kesulitan memahami prinsip kerja *marker beacon* karena keterbatasan fasilitas	Mengembangkan dan menghasilkan prototipe *marker beacon* berbasis Arduino Uno yang dapat mensimulasikan prinsip kerja peralatan tersebut	Jenis Penelitian: Research and Development (R&D) Model: ADDIE (Analysis, Design, Development,	Tidak disebutkan jumlah subjek manusia (responden), karena penelitian fokus pada uji alat dan kinerja prototipe. Pengujian dilakukan	Semua komponen bekerja sesuai desain: buzzer berbunyi sesuai nada morse, LED berkedip, sinyal terbaca di osiloskop. Deteksi sensor ultrasonik berhasil

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
		laboratorium. Pembelajaran hanya berupa teori tanpa media praktik yang konkret.	sebagai media pembelajaran yang konkret bagi Taruna/I.	Implementation, Evaluation) Tahapan: Analisis kebutuhan dan survei laboratorium Desain sistem prototipe (pemancar dan penerima) Pengembangan perangkat keras dan pemrograman Arduino Implementasi dan pengujian fungsi	terhadap 3 jenis *marker beacon*: • Outer Marker (400 Hz) • Middle Marker (1300 Hz) • Inner Marker (3000 Hz)	pada jarak yang berbeda: Outer (10 cm), Middle (7.5 cm), Inner (2.5 cm). Prototipe berhasil menampilkan sinyal audio sesuai standar marker beacon: • Outer Marker: 400 Hz • Middle Marker: 1300 Hz • Inner Marker: 3000 Hz Tampilan sinyal pada osiloskop mendukung validasi fungsi masing-masing

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
				Evaluasi efektivitas prototipe Peralatan: Arduino Uno, sensor ultrasonik, modul nRF24L01, buzzer, LED, generator sinyal AD9850, osiloskop		marker.
2	Media Pembelajaran Praja Muda Karana (Pramuka) Berbasis Android (Yusrina et al., 2024)	Proses pembelajaran pramuka di SMAN 5 Dumai masih menggunakan buku saku yang memiliki tampilan sederhana, ukuran kecil, berpotensi hilang, dan	Merancang dan membuat aplikasi pramuka berbasis android untuk memudahkan siswa dan guru dalam proses belajar mengajar.	Metode *Multimedia Development Life Cycle (MDLC)* dengan 6 tahap: konsep, desain, pengumpulan bahan, pembuatan, pengujian, dan	20 responden untuk pengujian aplikasi.	Aplikasi media pembelajaran pramuka berbasis android dengan materi video animasi, kuis tebak gambar, dan persentase kepuasan pengguna 91.7%

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
		membuat siswa kesulitan untuk membeli buku saku lagi.		distribusi.		(kategori sangat baik).
3	Sistem Pendeteksi Kemiripan Jawaban Essay Menggunakan Algoritma Rabin Karp (Siregar et al., 2022).	Banyaknya jawaban penalaran mahasiswa yang sama menyebabkan kurangnya kreatifitas pemikiran dari mahasiswa itu sendiri dan dosen memerlukan waktu pengoreksian yang cukup lama dan kurang efisien karena memeriksa jawaban mahasiswa dengan satu per satu.	Mendeteksi kemiripan jawaban essay dan membantu Dosen dalam memeriksa jawaban ujian essay mahasiswa satu dengan mahasiswa yang lainnya, serta meminimalisir untuk tidak saling mencontoh satu sama lain.	Algoritma Rabin Karp, dengan tahapan: Identifikasi Masalah, Pengumpulan Data, Analisis Kebutuhan, Analisis Perancangan Sistem, Implementasi, Pengolahan Data, dan Pengujian.	Dalam dokumen tersebut tidak disebutkan secara spesifik jumlah responden. Data yang digunakan adalah data jawaban essay mahasiswa.	Implementasi algoritma Rabin Karp berhasil mendeteksi kemiripan jawaban essay mahasiswa.
4	Implementasi Algoritma Rabin-	Maraknya plagiarisme antar	Membangun sistem berbasis	Menggunakan algoritma Rabin-	20 pasang dokumen	Sistem berhasil mengidentifikasi

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
	Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa (Filcha & Hayaty, 2019).	mahasiswa akibat kemudahan pengumpulan tugas secara digital dan kurangnya sistem deteksi otomatis.	web yang mampu mendeteksi tingkat kemiripan dokumen tugas mahasiswa menggunakan algoritma Rabin-Karp.	Karp dengan tahapan preprocessing teks (case folding, tokenizing, punctuation removal, stopword removal, stemming), dilanjutkan dengan pencocokan K-Gram dan hashing. Kemiripan diukur menggunakan Dice Coefficient, dan evaluasi akurasi dilakukan dengan confusion matrix.	mahasiswa diuji dan dibandingkan hasil sistem dengan software Plagiarism Checker X.	kemiripan dokumen dengan akurasi sebesar **90%**. Contoh hasil similarity antara dua dokumen adalah **78.05%**, tergolong *plagiarisme* berat. Tidak ada perbedaan hasil saat urutan perbandingan dokumen diubah.
5	Deteksi	Bagaimana cara	Tujuan dari	Metodologi	Jumlah data yang	Penelitian ini

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
	Plagiarisme Abstrak Skripsi dengan Menggunakan Algoritma Rabin Karp (Studi Kasus: Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang) (Indra Gunawan, 2021)	mendeteksi plagiarisme pada abstrak skripsi secara efektif dan efisien dengan menggunakan algoritma string matching Rabin-Karp?	penelitian ini adalah untuk membangun sistem deteksi plagiarisme pada abstrak skripsi dengan menerapkan algoritma Rabin-Karp sebagai metode pencocokan string guna mengetahui tingkat kemiripan teks secara efektif dan akurat.	penelitian ini dilakukan dengan studi literatur, pengumpulan data abstrak skripsi, pra-pemrosesan teks, implementasi algoritma Rabin-Karp untuk pencocokan string, serta evaluasi hasil deteksi plagiarisme berdasarkan tingkat kemiripan teks.	digunakan dalam penelitian ini adalah 100 data abstrak skripsi dari Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang.	berhasil menerapkan algoritma Rabin-Karp untuk mendeteksi plagiarisme pada abstrak skripsi dengan akurasi yang baik, meskipun diperlukan penggabungan dengan teknik lain untuk meningkatkan efektivitas deteksi pada teks yang sangat mirip.

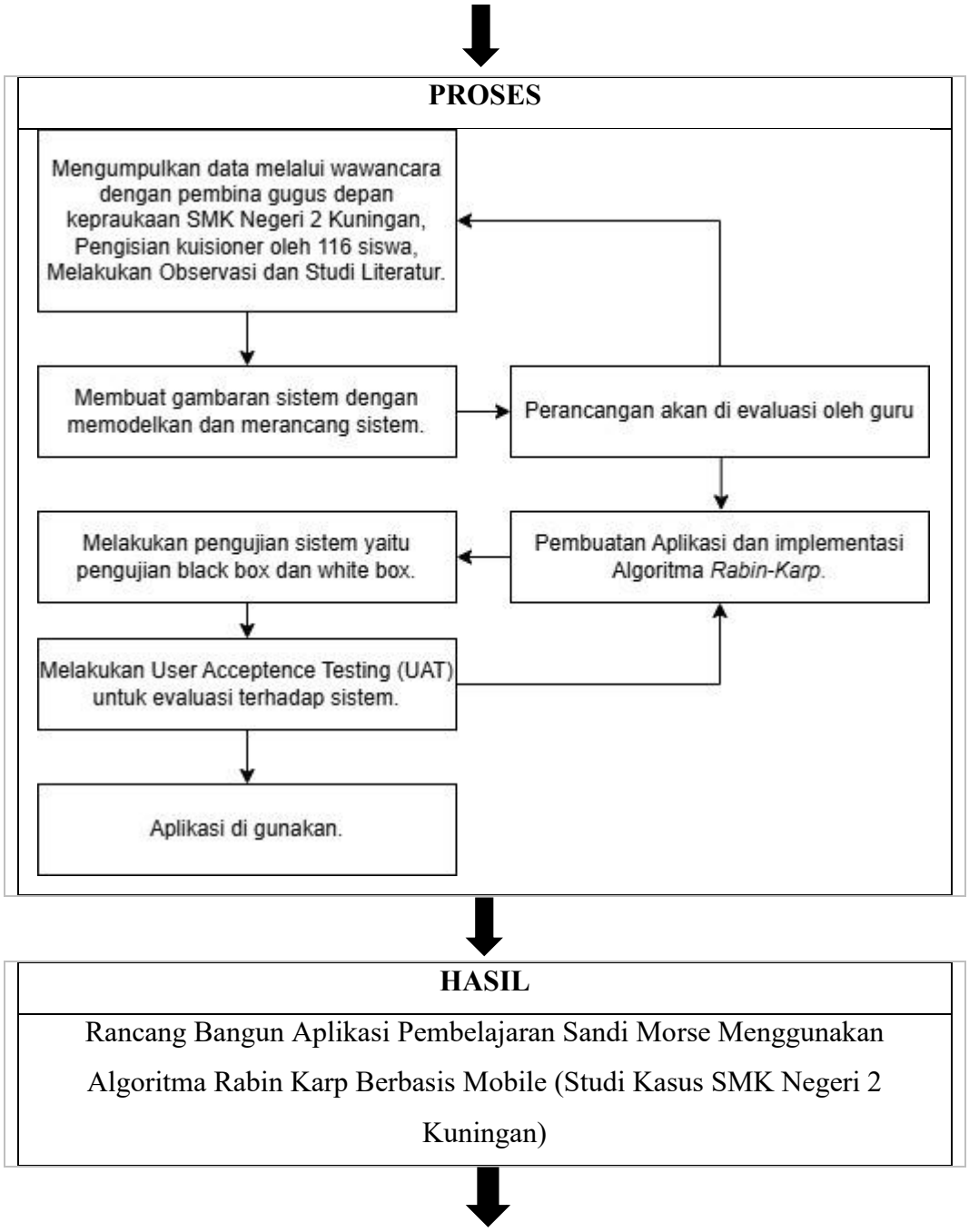
2.3 Kerangka Teoritis *(Theoritical Framework)*

Kerangka acuan yang digunakan oleh peneliti yang merupakan hasil sintesa dari teori-teori yang relevan dan analisis penelitian-penelitian terdahulu, yang kemudian dibandingkan sebagi kemungkinan-kemungkinan solusi bagi penelitian yang dilakukan.dapat dikembangkan sesuai dengan penelitian.

MASALAH
Di SMK Negeri 2 Kuningan, pembelajaran Sandi Morse masih mengandalkan buku teks dan metode ceramah sebagai bahan ajar utama. Pendekatan ini bersifat sepihak dan kurang melibatkan partisipasi aktif dari siswa, sehingga sering menimbulkan kejenuhan serta kesulitan dalam memahami materi. Dampaknya, minat dan motivasi siswa untuk belajar menjadi rendah, meskipun kemampuan dalam memahami Sandi Morse merupakan aspek penting dalam kegiatan Pramuka yang mendukung pelaksanaan Projek Penguatan Profil Pelajar Pancasila (P5). Selain itu, belum tersedia media pembelajaran yang memungkinkan siswa untuk belajar secara mandiri di luar kegiatan sekolah. Proses evaluasi pun masih dilakukan dengan cara observasi langsung oleh guru saat latihan berlangsung, yang menyulitkan dalam memantau perkembangan siswa secara menyeluruh. Penilaian jawaban siswa terhadap soal juga masih dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama, berpotensi terjadi kesalahan, dan menghambat pemberian umpan balik yang cepat kepada siswa.

⬇

TUJUAN
Membangun aplikasi pembelajaran sandi Morse berbasis mobile yang menggunakan algoritma Rabin-Karp untuk mencocokkan jawaban siswa secara otomatis, dengan tujuan meningkatkan pemahaman siswa terhadap materi sandi Morse secara lebih mudah, mandiri, dan dapat diakses kapan saja, serta mendukung pencapaian nilai-nilai dalam Profil Pelajar Pancasila.


PROSES
Mengumpulkan data melalui wawancara dengan pembina gugus depan kepraukaan SMK Negeri 2 Kuningan, Pengisian kuisioner oleh 116 siswa, Melakukan Observasi dan Studi Literatur.
Membuat gambaran sistem dengan memodelkan dan merancang sistem.
Perancangan akan di evaluasi oleh guru
Pembuatan Aplikasi dan implementasi Algoritma Rabin-Karp.
Melakukan pengujian sistem yaitu pengujian black box dan white box.
Melakukan User Acceptence Testing (UAT) untuk evaluasi terhadap sistem.
Aplikasi di gunakan.
HASIL
Rancang Bangun Aplikasi Pembelajaran Sandi Morse Menggunakan Algoritma Rabin Karp Berbasis Mobile (Studi Kasus SMK Negeri 2 Kuningan)

MANFAAT
Penelitian ini memberikan manfaat bagi siswa berupa media pembelajaran yang mudah digunakan, menarik, dan dapat diakses secara fleksibel untuk meningkatkan pemahaman terhadap sandi Morse; bagi sekolah, penelitian ini mendukung pelaksanaan kegiatan Pramuka dan penguatan Profil Pelajar Pancasila melalui pendekatan digital; bagi pembina atau guru, penelitian ini mempermudah proses evaluasi dan pemantauan pemahaman siswa; serta bagi peneliti atau pengembang, penelitian ini menjadi contoh penerapan algoritma Rabin-Karp secara praktis dalam bidang pendidikan.