

BAB II

TINJUAN PUSTAKA

2.1 Rancang Bangun

2.1.1 Definisi Rancang Bangun

Menurut (Hasyimetal.,2014) rancang bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Rancang bangun juga merupakan kegiatan menerjemahkan hasil Analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan *system* tersebut atau memperbaiki *system* yang sudah ada.

Menurut (T. Ramadhan & Utomo, 2014), rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah *system* kedalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen – komponen *system* diimplementasikan. Rancang bangun sangat berkaitan dengan perancangan *system* yang merupakan satu kesatuan untuk merancang dan membangun sebuah aplikasi.

Dari definisi tersebut, dapat ditarik Kesimpulan bahwa definisi rancang bangun adalah proses perencanaan yang menggambarkan urutan kegiatan (sistemmatika) mengenai suatu program atau aplikasi. Rancang bangun juga berkaitan dengan perancangan *system* Dimana rancang bangun sebagai penerjemah hasil analisa yang dibentuk ke dalam suatu software yang berguna untuk menciptakan *system* atau memperbaiki *system*.

2.1.2 Pengertian System

System adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan. Berkumpul bersama – sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto, 2005), penjelasan tersebut menjelaskan bahwa *system* bekerja dalam suatu jaringan kerja dari suatu prosedur nya yang saling berhubungan suatu sama lain untuk menyelesaikan tujuan dan sasaran yang dimaksud.

System juga diartikan sebagai sekumpulan elemen yang bekerja sama dalam suatu kesatuan untuk melaksanakan suatu fungsi yang berguna. Dalam bukunya Jogiyanto *system* dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen. Menurut (Kadir, 2017) adalah sekelompok elemen-elemen yang saling terintegrasi dengan maksud dan tujuan yang sama untuk melaksanakan sasaran yang telah ditentukan

2.1.3 Pengertian Informasi

Informasi menurut (Jogiyanto, 2005) adalah data yang diolah menjadi bentuk yang lebih berguna bagi penerimanya untuk mengambil keputusan saat ini atau saat mendatang. Penjelasan tersebut bahwa informasi juga memiliki peran yang sangat penting dalam mengambil keputusan karena suatu informasi yang akan disampaikan harus mengenai sasaran yang telah ditentukan.

2.1.4 Pengertian System Informasi

System informasi adalah suatu *system* dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. *System* informasi (Kadir, 2017) adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, *computer*) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sasaran Perusahaan.

2.1.5 Komponen System Informasi

1. *System* informasi manajemen: *System* perencanaan yang melibatkan bagian internal Perusahaan yang meliputi pemanfaatan teknologi, prosedur, dan interaksi manusia untuk memecahkan permasalahan bisnis seperti layanan, biaya produksi, atau penentuan strategi bisnis yang tepat.
2. *System* pendukung keputusan: salah satu bagian dalam informasi berbasis computer yang berfungsi untuk mengambil sebuah keputusan yang tepat bagi Perusahaan, organisasi, atau instansi terkait.

3. *System* informasi eksekutif: salah satu jenis dari *system* informasi manajemen untuk memudahkan dalam mengambil kebijakan yang dibutuhkan oleh eksekutif senior dengan menampilkan akses informasi yang tepat dan relevan.
4. *System* pemrosesan transaksi: bagian dari *System* informasi yang memiliki tugas untuk menjalankan dan mencatat setiap kebutuhan transaksi secara berkelanjutan (*continue*) yang diperlukan dalam bisnis.

2.1.6 Fungsi *System* Informasi

Ada beberapa fungsi *system* informasi di antara nya:

1. Mampu menjadi Tingkat ketersediaan kualitas dan pengalaman dalam mengelola sebuah *system* informasi secara kritis dan logis.
2. Mampu meningkatkan produktivitas kerja pada seluruh stakeholder.
3. Mampu menganalisa dan meminimalisir terjadinya kerugian dari sisi ekonomi.
4. Memberikan aksesibilitas yang baik kepada pengguna.
5. Dapat mencapai tujuan perusahaan secara cepat berdasarkan dukungan data yang dapat dipertanggung jawabkan.

2.1.7 *System* Informasi Berbasis *Web Mobile*

Menurut Menurut (Safaat, 2012) *Mobile Web* merupakan sebuah kumpulan halaman *HTML* dengan berbasis browser. Yang dapat diakses dengan menggunakan perangkat portable seperti *smartphone* dan *gadget/tablet*. Melalui jaringan internet atau telekomunikasi berupa 3G, 4G, atau wifi. *Mobile Web* ini dirancang untuk menampilkan sebuah text atau konten data gambar, audio dan video atau penggabungan dari beberapa konten yang berada pada layar sebuah *smartphone* dan tentu juga di pengaruhi oleh keterbatasan ukuran layar dan jenis layar sentuhnya.

2.2 Pramuka dan Pramuka Penggalang

2.2.1 Pengertian Pramuka

Pramuka adalah singkatan dari Praja Muda Karana (berarti orang-orang berjiwa muda yang suka berkarya), yaitu setiap pemuda yang aktif dalam kegiatan kepramukaan dan didik dengan berbagai keterampilan,

disiplin, kepercayaan pada diri sendiri, saling menolong, dan lain sebagainya (Nasruddin, 2018).

Pramuka Penggalang adalah salah satu tingkatan dalam Gerakan Pramuka yang ditujukan bagi anak-anak usia 11-15 tahun. Pramuka Penggalang fokus pada pembentukan karakter melalui kegiatan pendidikan, kepramukaan, dan pengabdian kepada masyarakat.

2.2.2 Lagu-lagu dan Yel-yel Pramuka

Lagu-lagu dan yel-yel pramuka adalah sebuah nada atau sorakan yang digunakan untuk menyenangkan anggota pramuka baik dalam suasana senang maupun sedih.

2.2.3 Tanda/Atribut Pramuka

Tanda pengenal pramuka adalah tanda yang dikenakan oleh seorang pramuka pada seragam yang menunjukkan jati dirinya sebagai seorang pramuka, satuan tempatnya bergabung, jabatan yang diembannya, kemampuan dan kecakapannya, dan penghargaan yang telah diterimanya (Nasruddin, 2018). Dalam gerakan pramuka, tanda pengenal terdiri dari:

1. Tanda Umum



Gambar 2. 1 Tanda Umum Pramuka

<https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEj4HbxtPvcdFJWqbTVO39y0rgGrnPr1G-DJrMxWDCOhtRfB0QMZFyRGtpBpueA5-67omu35pkSVeZRIixQ9gCmNrKd4871icTdtqTM-jTgdbbTse3IrcCFENpUXtdPq3Ms2ea0qGVccQ3o/s320/tanda-tutup-kepala.jpg>

Tanda Umum Pramuka adalah tanda pengenal yang dikenakan oleh semua anggota pramuka pada kegiatan seragamnya dan gerakan pramuka sedunia. Tanda umum berfungsi sebagai jati diri seorang anggota pramuka.

2. Tanda Satuan

Tanda Satuan adalah kelompok tanda pengenal dalam gerakan pramuka yang menunjukkan seorang anggota Gerakan Pramuka tergabung dalam satuan atau kwartir tertentu. Satuan atau kwartir di sini mulai dari satuan terkecil (barung, regu, dan krida), gugusdepan, kwartir ranting, hingga kwartir nasional. Tanda satuan dipasang di pakaian seragam pramuka. Penggunaan tanda satuan ini dimaksudkan untuk mempermudah mengenal satuan wilayah tempat anggota Gerakan Pramuka bergabung.

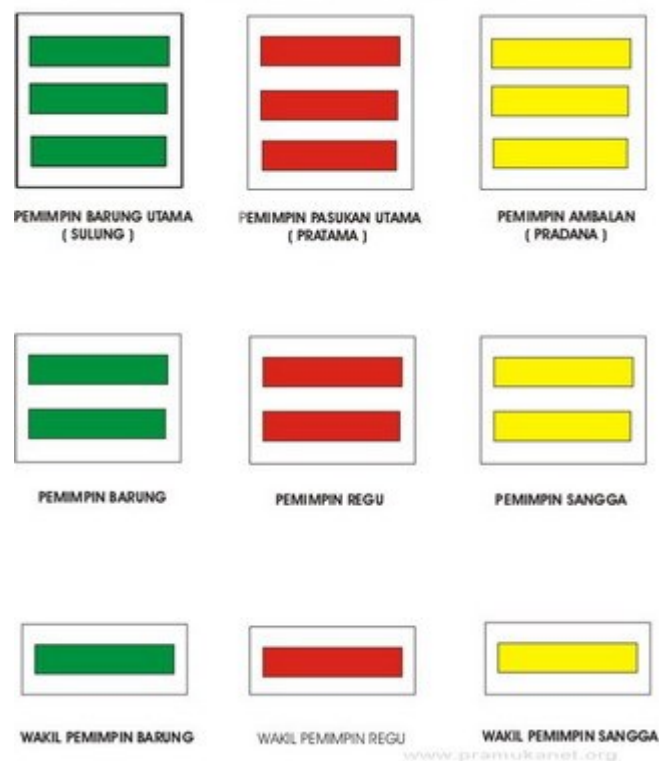


Gambar 2. 2 Tanda Lencana Wilayah

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRixSt4eJ2QJTue-9_QTy_oQvCoJuzn2RTiXQ&s

3. Tanda Jabatan

Tanda Jabatan adalah tanda pengenal pramuka yang menunjukkan jabatan seorang beserta hak dan kewajiban yang melekat dengan jabatan itu.



Gambar 2. 3 Tanda Jabatan
<https://pramukanet.id/wp-content/uploads/2016/10/t.jab-pinpesdik.jpg>

4. Tanda Kecakapan

Tanda Kecakapan dalam Gerakan Pramuka adalah tanda-tanda yang dikenakan pada pakaian pramuka yang menunjukkan kecakapan, keterampilan, ketangkasan, kemampuan, sikap dan usaha seorang Pramuka dalam bidang tertentu, sesuai dengan golongan usiannya. Secara singkat tanda kecakapan bisa diartikan sebagai tanda yang menunjukkan kecakapan atau kemampuan seorang pramuka pada bidang keterampilan tertentu.



Gambar 2. 4 Tanda Kecakapan

https://blue.kumparan.com/image/upload/f1_progressive,fl_lossy,c_fill,f_auto,q_auto:best,w_640/v1637224729/souzyrozk1lwcp_yhrig.jpg

2.2.4 Tali-Temali

Sebagai alat bantu, tali banyak dipakai untuk mengikat, mengangkat, atau menarik sebuah benda, tali-temali atau knots merupakan salah satu bagian dari kehidupan sehari-hari, mulai dari tiang jemuran, menali kapal, membuat kemah hingga panjat tebing, dalam kegiatan alam terbuka (*Outdoor*), pengetahuan tali-temali menjadi pengetahuan wajib yang harus dimiliki oleh penggiat alam terbuka, bukan hanya untuk diri sendiri, tetapi juga berguna untuk ketika berada dalam keadaan darurat, baik saat mendaki gunung maupun panjat tebing (Nasruddin, 2018).



Gambar 2. 5 Kumpulan Simpul
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4-yB6I1vO5NYdkmTP9aL1b0Oj1D_QK2toCg&s

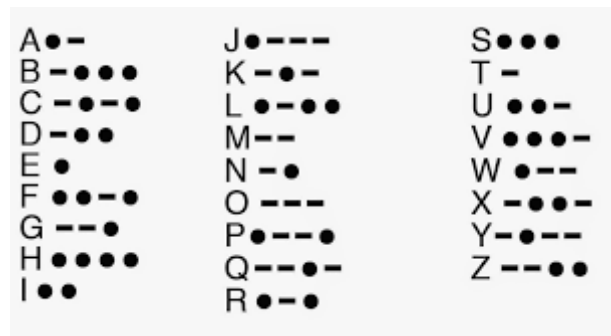
2.2.5 Sandi-sandi

Sandi berasal dari bahasa Sanskerta yang berarti rahasia atau tulisan-tulisan yang dirahasiakan, disebut rahasia karena huruf atau sandi hanya dapat dimengerti oleh orang yang mengetahui kata kuncinya.

Beberapa contoh sandi – sandi yang ada di dalam pramuka sebagai berikut:

1. Sandi Morse

Sandi morse adalah sandia tau kode komunikasi menggunakan suara sebagai pengantarnya, bisa melalui siulan, peluit, maupun Cahaya



Gambar 2. 6 Sandi Morse

<https://awsimages.detik.net.id/community/media/visual/2022/12/16/sandi-morse-pramukadiyoriid-1.jpeg?w=1163>

2. Sandi Kotak 1

Sandi kotak 1 pada dasarnya adalah sandi dengan penulisan menyerupai garis lekuk seperti gambar 2.7 dibawah ini.

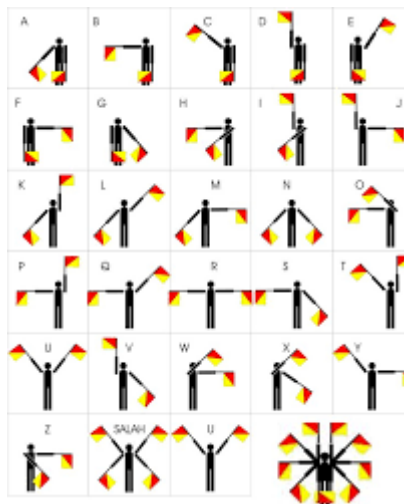
SANDI KOTAK I :



Gambar 2. 7 Sandi Kotak 1

<https://static.promediateknologi.id/crop/0x0:0x0/0x0/Webp/photo/p2/01/2024/08/22/Sandi-Kotak-1-2671816065.jpg>

Berbeda dengan semaphore. Semaphore atau semafor adalah salah satu Teknik untuk mengirim atau menyampaikan dan menerima pesan atau informasi dengan jarak yang berjauhan dengan menggunakan kombinasi kedua tangan. Agar mudah dilihat, kedua tangan biasanya memegang sebuah benda yang mencolok seperti tongkat, dayung, bendera, atau yang lainnya (Nasruddin, 2018).



Gambar 2. 7 Semaphore
<https://i.pinimg.com/736x/7c/20/a5/7c20a58a6bda1e1b026150d3b3477175.jpg>

2.3 Website

2.3.1 Definisi Website

Menurut (Heni, 2010), *Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia, selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau Kumpulan komponen yang terdiri dari teks, gambar, suara dan animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi, secara garis besar *website* bisa digolongkan menjadi 4 jenis yaitu:

1. *Website* statis adalah *web* yang mempunyai halaman tidak berubah. Artinya, untuk melakukan perubahan pada suatu halaman dilakukan. Secara manual dengan mengedit kode yang menjadi struktur dari *website* tersebut.
2. Secara manual dengan mengedit kode yang menjadi struktur dari *website* tersebut.
3. *Website* dinamis adalah *website* yang secara struktur dipermudahkan melakukan update seiring mungkin, *user* dapat memperbarui data pada halaman *web*.

4. *Website* interaktif, pada *web* ini *user* beradu argument mengenai apa yang menjadi pemikiran mereka. Contoh *website* adalah blog dan forum.

2.3.2 Definisi *Website Mobile*

Web mobile merupakan *web* atau halaman *website* internet yang dapat digunakan atau diakses pada perangkat mobile. *Mobile web* merupakan platform yang paling mudah untuk dipelajari, paling murah untuk diproduksi, terstandarisasi, yang paling tersedia, dan paling mudah untuk didistribusikan sesuai dengan prinsip *Ubiquity*. *Web* mobile adalah *web* atau halaman *website* yang dapat digunakan atau diakses pada perangkat mobile serta mudah untuk dipelajari, paling murah, dan paling mudah untuk didistribusikan (Faulina, 2016).

2.4 *Apache*

Apache adalah salah satu aplikasi *web server*, *apache* bertugas menerjemahkan *Uniform Resource Locator* (URL) menuju file, kemudian mengirimkan file tersebut melalui internet, atau ke program yang kemudian dijalankan oleh program tersebut dan mengirimkan hasilnya (Laurie & Laurie, 2003). Kelebihan-kelebihan *Apache* menurut (Kabir, 2002) adalah:

1. *Apache is highly configurable web server with a modular design* (didesain dengan *system* modular dan dapat dikonfigurasi).
2. *Apache is free, open source techonolgy* (gratis, dan *souce code* diberikan secara bebas).
3. *Apache works great with Pearl, PHP, and other scripting language* (dapat bekerja dengan *Pearl*, *PHP*, dan Bahasa scripting lain).
4. *Apache runs on Linux and other Unix Systems* (dapat berjalan Linux dan *System* Unix lainnya).
5. *Apache also runs on Windows* (dapat berjalan pada Microsoft Windows).

2.5 *PhpMyAdmin*

PhpMyAdmin adalah sebuah aplikasi atau perangkat berbasis open source yang bisa kita gunakan secara gratis untuk melakukan pemograman

ataupun Administrasi pada database *MySQL*. *PhpMyAdmin* sendiri menggunakan Bahasa *PHP* untuk pemogramannya (Sumantri et al., 2022), selain itu *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*realtions*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain (R. F. Ramadhan & Mukhaiyar, 2020)

2.6 *MySQL*

MySQL merupakan Database Management System (DBMS) tools open source yang mendukung multi *user*, *multithreaded*, popular, dan *free*. Berdasarkan teori diatas maka dapat disimpulkan bahwa SQL adalah bahasa permintaan database tertentu dimana sub bahasa dapat membuat dan memanipulasi data di dalam database. *SQL* digunakan untuk melakukan tugas-tugas seperti melakukan update terhadap database, yang merujuk pada konsep Realtional Database Management System (RDBMS) (Noviana, 2022).

2.7 *PHP : Hypertext Preprocessor*

PHP singkatan dari *Hypertext Preprocessor* adalah bahasa pemograman yang memungkinkan para *WEB Developer* untuk membuat aplikasi *Web* yang dinamis dengan cepat *PHP* merupakan singkatan dari “*PHP*” : *Hypertext Preprocessor*. *PHP* ditulis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya yang digunakan untuk mengetahui siapa saja yang telah mengakses ringkasan online-nya menurut (Kustiyahningsih & Anamisa, 2011), “*PHP* atay *Hypertext Preprocessor* adalah skrip bersifat *server-side* yang ditambahkan ke dalam *HTML*.”

PHP mengenal beberapa jenis tipe data yaitu: *integer*, *floating point*, *string*, *arrays*, *object*. Tipe data pada suatu variable secara otomatis akan ditentukan *PHP* tergantung pada operasi yang sedang berlangsung.

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode *PHP* untuk mengirimkan permintaan ke *server*, *System* kerja dari *PHP* diawali dengan

permintaan yang berasal dari halaman *Website* oleh *browser*. Berdasarkan *URL* atau alamat *website* dalam jaringan *internet*, *browser* akan menemukan sebuah alamat dari *Webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *Webserver*.

Selanjutnya *Webserver* akan mencari berkas yang diminta dan menyampaikan isinya browser, browser yang mendapatkan isinya segera menjermahkan kode *HTML* dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh *user* adalah halaman yang mengandung script *PHP*. Pada prinsipnya sama dengan memanggil kode *HTML*, namun pada saat permintaan dikirim ke *Web-server*, *Webserver* akan memeriksa tipe file yang diminta *user*, jika tipe file yang minta adalah *PHP*, maka akan memeriksa isi script dari halaman *PHP* tersebut.

Apabila dalam file tersebut tidak mengandung script *PHP* permintaan *user* akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung script *PHP*, maka proses akan dilanjutkan ke modul *PHP* sebagai mesin menerjemahkan script-script *PHP* dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode *HTML*, lalu ditampilkan ke browser *user*. (Kustiyangningsih, 2016).

2.8 UML (*Unified Modeling Language*)

Unified Modeling Language menurut Rudy (2012: 149) Bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun *System* software dari Bahasa pemodelan berorientasi objek

Menurut (Nugroho, 2010) Bahasa pemodelan untuk *system* atau software yang mempunyai paradigma berorientasi objek. Pemodelan (*modeling*) yang sebenarnya digunakan untuk menyederhanakan permasalahan – permasalahan yang kompleks dengan sedemikian rupa sehingga lebih mudah untuk mempelajarinya dan memahaminya.

Unified Modeling Language (UML) sendiri juga merupakan standar penulisan sebuah *System blue print*, yang meliputi konsep dalam proses bisnis, penulisan sebuah class dalam Bahasa pemograman yang spesifikasi,

skema dalam basis data, dan komponen lainnya yang diperlukan dalam *system*. *UML* diaplikasikan untuk maksud tertentu, biasanya diantara lain:

1. Merancang sebuah software
2. Sebagai sarana komunikasi antara sebuah software dengan bisnis proses.
3. Menjabarkan sebuah *system* secara lebih detail untuk menganalisa dan mencari apa yang diperlukan oleh *system*.
4. Membuat dokumentasi *system* yang sudah ada. Proses dan organisasinya.

Blok pada pembangunan utama sebuah *UML* yaitu diagram. Berapa diagram, ada yang lebih detail (seperti jenis timing diagram) dan ada juga yang bersifat umum seperti (diagram kelas). Untuk menggambarkan, membangun dan mendokumentasikan *system* yang dirancang para pengembang menggunakan sebuah *system* berorientasi objek dengan menggunakan bahasa model. *UML* juga memungkinkan para anggota team untuk bekerja sama satu sama lain dengan bahasa model yang sama dalam menerapkan beragam *system*. Oleh karena itu *UML* adalah sebuah alat komunikasi yang konsisten dalam mendukung para pengembang *system* saat ini. Sebagai perancang sekedar membaca diagram *UML* buatan orang lain.

Beberapa literatur menyebutkan *UML* menyediakan Sembilan jenis diagram. Akan tetapi kesembilan diagram ini tidak mutlak harus diterapkan dalam pengemabangan software, dikarenakan semuanya dibuat sesuai dengan kebutuhan. Beberapa diagram yang sering digunakan yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram*.

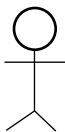
2.8.1 Use Case Diagram

Use Case Diagram merupakan suatu diagram yang menangkap kebutuhan bisnis untuk *system* dan untuk menggambarkan interaksi antara *system* dan lingkungan nya. (Tegarden et al., 2012), Dari kutipan tersebut dapat disimpulkan *Use Case Diagram* adalah suatu pemodelan *system*, aspek penting adalah menangkap perilaku dinamis. Perilaku dinamis berarti perilaku *system* saat sedang berjalan / beroperasi.

Hanya perilaku statis tidak cukup untuk memodelkan *system*, perilaku dinamis lebih penting dari pada perilaku statis. Dalam *UML*, tersedia lima diagram untuk memodelkan sifat dinamis dan *Use Case Diagram* salah satunya. Sekarang kita harus membahas bahwa diagram *Use Case* bersifat dinamis, harus ada beberapa factor internal atau eksternal untuk membuat interaksi.

Diagram *Use Case* terdiri dari *actor*, *Use Case*, dan hubungannya. Diagram digunakan untuk memodelkan *system*/subsistem dari suatu aplikasi. Diagram kasus penggunaan tunggal menangkap fungsionalitas tertentu dari suatu *system*. Oleh karena itu untuk memodelkan seluruh *system*, sejumlah diagram *Use Case* digunakan.

Tabel 2.1 Use Case

Simbol	Nama	Keterangan
	<i>Actor</i>	Kesatuan eksternal yang berinteraksi dengan <i>system</i> .
	<i>Use Case</i>	Rangkaian/uraian sekelompok yang saling terkait dan membentuk <i>system</i> .
	<i>Generelation</i>	Hubungan khusus atau interaksi antar objek.

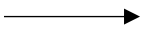
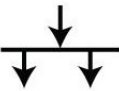
2.8.2 Activity Diagram

Activity Diagram menggambarkan alur kerja bisnis independent dari *class*, aliran kegiatan dalam *Use Case*, atau desain rinci sebuah metode. (Tegarden et al., 2012) *Activity Diagram* adalah diagram penting lainnya dalam *UML* untuk menggambarkan aspek dinamis dari *system*. Diagram aktivitas pada dasarnya merupakan diagram alir untuk mempersentasikan

aliran dari suatu aktivitas ke aktivitas lainnya. Aktivitas tersebut dapat digambarkan sebagai operasi *system*.

Aliran kendali ditarik dari satu operasi ke operasi lainnya. Aliran ini bisa beruntun, bercabang atau bersamaan. Diagram aktivitas menangani semua jenis control aliran dengan menggunakan elemen yang berbeda seperti garpu, gabungan, dan lainnya.

Tabel 2.2 Activity Diagram

Simbol	Nama	Keterangan
	<i>Star State</i>	Kondisi awal sebuah objek sebelum ada perubahan keadaan.
	<i>End State</i>	Menggambarkan Ketika objek objek berhenti memberi respon terhadap sebuah event.
	<i>State/Activities</i>	Menggambarkan kondisi sebuah entitas dan digambarkan dengan segi empat yang memiliki sudut tumpul
	<i>Line Conector</i>	Digunakan untuk menghubungkan satu symbol dengan symbol lainnya.
	<i>Fork</i> (Percabangan)	Digunakan untuk menunjukan kegiatan yang dilakukan secara paralel.
	<i>Join</i>	Digunakan untuk menunjukan kegiatan yang digabungkan.

	<i>Descision</i>	Digunakan untuk menggambarkan suatu keputusan / Tindakan yang harus diambil pada kondisi tertentu.
---	------------------	--

2.8.3 Class Diagram

Class Diagram adalah ilustrasi hubungan antara *class* yang dimodelkan di dalam *system*. *Class Diagram* sangat mirip dengan diagram hubungan entitas (ERD). *Class Diagram* menggambarkan *class* yang meliputi atribut, perilaku dan status, sementara dama ERD hanya mencangkup atribut (Tegarden et al., 2012).

Class Diagram adalah diagram statis. Diagram ini dapat mewakili sebuah tampilan statis dalam suatu aplikasi. *Class Diagram* digunakan juga untuk membangun kode yang dapat dieksekusi dari aplikasi dan tidak hanya digunakan untuk memvisualisasikan, mendeskripsikan, dan mendokumentasikan berbagai aspek pada sebuah *system*. Diagram kelas menggambarkan atribut dan operasi kelas dan juga Batasan yang dikenakan pada *system*. Diagram kelas banyak digunakan dalam pemodelan *system* berorientasi objek karena hanya diagram *UML*, yang dapat dipetakan langsung dengan bahasa berorientasi objek. Diagram kelas menunjukkan kumpulan kelas, antar muka, asosiasi, kolaborasi, dan batasan. Ini juga dikenal sebagai structural.

Tabel 2.3 Class Diagram

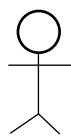
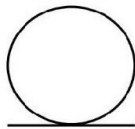
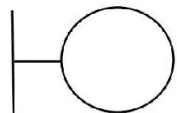
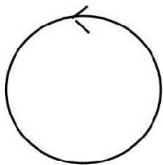
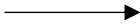
Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atanya objek induk (<i>ancestor</i>).
	<i>Nary Associanto</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan <i>System</i> yang menghasilkan suatu hasil yang terstruktur bagi suatu actor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.8.4 Sequence Diagram

Sequence Diagram merupakan urutan model dinamis yang menggambarkan contoh *class* yang ada dalam *Use Case* dan pesan yang lewat diantara mereka dari waktu ke waktu (Tegarden et al., 2012). *Sequence Diagram* merupakan diagram interaksi yang tersusun dengan urutan waktu. Cara membaca sebuah diagram sekuensial yaitu dari atas ke bawah. Setiap diagram sekuensial dapat mempersentasikan satu *flow* dari beberapa *flow* di dalam *Use Case*.

Tabel 2.4 Squence Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang/perangkat keras/ <i>System</i> lain yang sedang berinteraksi dengan <i>System</i> .
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form.
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan table.
	<i>A focus of control & a life line</i>	Menggambarkan tempat mulai dan berakhirnya sebuah message.
	<i>A Message</i>	Menggambarkan pengiriman pesan.

2.9 Blackbox Testing

Blackbox Testing atau biasa disebut juga dengan nama *Behavioral Test* merupakan sebuah pengujian yang dilakukan dengan cara mengamati

hasil *input* dan *output* dari sebuah perangkat lunak tanpa harus mengetahui struktur kode dari perangkat lunak tersebut (Nidhra & Dondeti, 2012).

Biasanya pengujian ini dilakukan di akhir proses pembuatan perangkat lunak sebagai cara untuk mengetahui apakah perangkat lunak tersebut bisa berfungsi dengan baik. Akan tetapi proses pengujian juga bisa dilakukan walaupun perangkat lunak baru selesai Sebagian.

2.9.1 Jenis Teknik *Blackbox Testing*

Dalam *Blackbox Testing* ada beberapa jenis yang biasanya digunakan untuk menguji sebuah perangkat lunak (Nidhra & Dondeti, 2012), berikut dibawah adalah beberapa tekniknyanya sebagai berikut:

1. *All Pair Testing*

Teknik ini juga bisa disebut sebagai teknik *pairwise testing*, pengujiannya digunakan untuk menguji semua kemungkinan dari kombinasi seluruh hubungan berdasarkan parameter input.

2. *Boundary Value Analysis*

Teknik ini focus kepada pencarian error dari dala atau sisi luar perangkat lunak.

3. *Cause-effect Graph*

Pada Teknik ini menggunakan grafik sebagai patokan, grafik tersebut menggambarkan relasi antara efek dan penyebab dari error di perangkat lunak.

4. *Equivalence Partitioning*

Teknik ini pengujian ini membagi data input dari beberapa peranti lunak menjadi beberapa bagian partisi data.

5. *Fuzzing*

Teknik ini merupakan Teknik pencarian bug di dalam peranti lunak dengan cara memasukan data yang salah atau tidak sempurna.

6. *Orthogonal Array Testing*

Teknik ini biasanya digunakan jika terdapat input yang kecil, tapi cukup bersifat bila digunakan dalam skala besar.

7. *State Transition*

Teknik ini memiliki keunggulan untuk melakukan pengujian kepada mesin dan navigasi dari antarmuka dalam bentuk grafik.

2.10 *SDLC (Software Development Life Cycle)*

SDLC dimulai dari tahun 1960-an, untuk mengembangkan *system* skala usaha besar secara fungsional untuk pada konglomerat pada zaman itu. *System* – *system* yang dibangun mengelola informasi kegiatan dan rutinitas dari perusahaan – perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya.

SDLC (Software Development Life Cycle) adalah proses mengembangkan atau merubah suatu *System* perangkat lunak dengan menggunakan model – model dan metodologi yang digunakan untuk mengembangkan *system* – *system* perangkat lunak sebelumnya (Shalahuddin & Rosa, 2013)

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Namun setiap prosesnya yang terutama adalah mengetahui keinginan tipe pelanggan dan memilih model *SDLC* sesuai keinginan pelanggan serta sesuai dengan karakteristik pengembangan *system*. Ada beberapa model *SDLC* menurut (Shalahuddin & Rosa, 2013) diantaranya:

1. *Model Waterfall*

Model air terjun (*waterfall*) ini juga disebut model sekuensial linear (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model ini menyajikan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahapan pendukung (*support*).

2. *Model Prototipe*

Model *prototipe* dapat digunakan untuk menyambungkan ketidakpastian pelanggan mengenai hak teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Program protipe biasanya merupakan program yang belum jadi.

3. *Model Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model *RAD* adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

4. Model *Iteratif*

Model Iteratif (*Iteratif model*) mengkombinasikan proses – proses oada air terjun dan iterative. Pada model prototipe model inkrementak akan menghasilkan versi – versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen/increment).

5. Model *Spiral*

Model spiral (spiral model) memasangkan iterative pada model prototipe dengan control dan aspek *Systematik* yang diambil dari model air terjun.

2.11 *Rapid Application Development*

2.11.1 Pengertian *Rapid Application Development*

Rapid Application Development (RAD) merupakan merode pembuatan desain aplikasi dengan pendekatan yang digunakan pada proses pembuatannya menggunakan desain contoh seperti prototipe, dengan pengembangan yang memanfaatkan umpan balik dari para penggunanya, yang langsung dieksekusi dan diperbaiki Ketika ditemukan masalah yang terjadi Ketika proses pembuatannya (Hassan et al., 2015)

Metode ini berfokus pada pembuatan prototipe secara cepat dan mengandalkan feedback dari *User*. Keuntungan utama menjalankan RAD adalah jangka watu pengembangan lebih cepat. Hal ini dikarenakan feedback dari pengguna *System* cepat didapatkan dan semua perubahan yang dilakukan sesuai hasil tersebut.

2.11.2 Kelebihan *Rapid Application Development*

Penggunaan *Rapid Application Development* (RAD) dalam merancang sebuah *System* memiliki beberapa kelebihan dalam merancang

sebuah *System* (Wahyuningrum & Januarita, 2014), kelebihan RAD sebagai berikut :

1. Memerlukan kerja sama tim yang memiliki kemampuan yang baik.
2. Membutuhkan kerja sama tim yang kuat.
3. Tidak berlaku untuk proyek yang Panjang.
4. Focus pada penggunaan aplikasi yang modular.
5. Sukar jika dibuat untuk aplikasi yang besar dan kompleks.

2.11.3 Kekurangan *Rapid Application Development*

Penggunaan *Rapid Application Development* (RAD) dalam merancang sebuah *system* yang tidak lepas dari kekurangan dalam merancang sebuah *system* (Wahyuningrum & Januarita, 2014), kekurangan RAD sebagai berikut :

1. Kriteria aplikasi dapat berubah setiap saat.
2. Mengembangkan aplikasi sesuai dengan kebutuhan pengguna.
3. Menimalisir error dan masalah lain.
4. Dapat membuat desain *system* dengan cepat dan efektif.
5. Mudah dalam membuat proses integrasi.

2.12 Kajian Penelitian yang Relavan

Ada beberapa hasil penelitian terdahulu yang relavan atau berhubungan dengan penelitian yang dilakukan oleh penulis, yaitu sebagai berikut :

1. Penelitian yang dilakukan oleh Hamdani dan Hendriyani, dengan berjudul “Perancangan Media Pembelajaran Gerakan Semaphore Pramuka Berbasis Augmented Reality Dengan Marker Based Tracking. Algoritma: Jurnal Ilmu Komputer dan Informatika.”

Persamaan: memiliki tujuan yang sama yaitu berfokus pada pramuka penggalang dengan berbagi aspek seperti pembelajaran, evaluasi program.

Perbedaan: berfokus pada pengembangan media pembelajaran interaktif untuk gerakan semaphore dalam kepramukaan menggunakan

Augmented Reality (AR) dengan Marker-Based Tracking. Aplikasi yang mereka rancang memungkinkan pengguna untuk melihat simulasi gerakan semaphore dalam bentuk visual 3D, sehingga meningkatkan pemahaman dan praktik anggota Pramuka Penggalang dalam menggunakan kode semaphore.

Di sisi lain, penelitian dengan judul "Rancang Bangun *System* Informasi Panduan Buku Pramuka Penggalang di Kwarda Garut Berbasis *Web*" lebih berfokus pada pengembangan *System* informasi berbasis *Web* yang menyediakan akses digital terhadap materi buku panduan pramuka, membantu anggota Pramuka dalam memahami berbagai materi yang terdapat dalam buku pegangan mereka.

Meskipun kedua penelitian ini bertujuan untuk meningkatkan pembelajaran kepramukaan, metode yang digunakan sangat berbeda. Hamdani & Hendriyani (2022) menggunakan teknologi AR untuk menciptakan pembelajaran visual dan interaktif, sedangkan penelitian *System* informasi berbasis *Web* lebih menitikberatkan pada aksesibilitas informasi melalui platform digital. Dengan demikian, penelitian pertama lebih berfokus pada pembelajaran keterampilan semaphore, sedangkan penelitian kedua lebih luas cakupannya dalam penyediaan materi pramuka secara digital.

2. Penelitian yang dilakukan oleh Sari, Dwi dan Rahmat dengan berjudul "Pengembangan *System* Informasi Buku Saku Digital Pramuka Berbasis Android untuk Pramuka Penggalang."

Persamaan: kedua penelitian memiliki tujuan yang sama, yaitu meningkatkan akses dan pemahaman materi kepramukaan bagi anggota Pramuka Penggalang melalui pemanfaatan teknologi informasi. Keduanya juga menitikberatkan pada digitalisasi materi pramuka agar lebih mudah diakses oleh anggota, baik untuk keperluan belajar mandiri maupun kegiatan kelompok.

Perbedaan: penelitian yang dilakukan oleh Sari, Dwi, dan Rahmat berfokus pada pengembangan aplikasi buku saku digital berbasis android yang dapat diunduh dan digunakan secara offline diperangkat

mobile. Aplikasi ini menyediakan berbagai materi kepramukaan dalam bentuk ringkasan, gambar, dan kuis interaktif untuk mendukung proses pembelajaran anggota Pramuka Penggalang.

Sementara itu, penelitian dengan judul “Rancang Bangun *System* Informasi Buku Panduan Pramuka Tingkat Penggalang Berbasis *Web* Kwarcab Garut” lebih menitikberatkan pada pengembangan *System* informasi berbasis *Web* yang dapat diakses secara online oleh seluruh anggota pramuka Kwarcab Garut. *System* ini menyediakan akses digital terhadap buku panduan pramuka secara lengkap, serta fitur – fitur pencarian dan pengelolaan data anggota.

3. Penelitian oleh Lestari, N., & Pratama, R. (2021), berjudul “Rancang Bangun E-Learning Pramuka Berbasis *Web* untuk Mendukung Kegiatan Pramuka di Sekolah Dasar.” *Jurnal Pendidikan dan Teknologi Informasi.*”

Persamaan: Kedua penelitian bertujuan untuk meningkatkan efektivitas pembelajaran kepramukaan melalui pemanfaatan teknologi berbasis *Web*. Sama-sama menyediakan materi, panduan, dan fitur interaktif yang dapat diakses oleh anggota Pramuka Penggalang, sehingga mendukung proses belajar yang lebih fleksibel dan modern.

Perbedaan: Penelitian yang dilakukan oleh Lestari dan Pratama berfokus pada pengembangan platform e-learning berbasis *Web* yang tidak hanya menyediakan materi, tetapi juga fitur evaluasi seperti kuis online, forum diskusi, dan penilaian otomatis untuk siswa sekolah dasar yang mengikuti kegiatan pramuka. Sebaliknya, penelitian "Rancang Bangun *System* Informasi Buku Panduan Pramuka Tingkat Penggalang Berbasis *Web* Kwarcab Garut" lebih mengedepankan digitalisasi buku panduan pramuka secara menyeluruh, dengan fitur utama berupa akses ke seluruh isi buku panduan, pengelolaan data anggota, dan pencatatan aktivitas pramuka.

2.13 Kerangka Berpikir

Gambar 2. 8 Kerangka Berpikir

