

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

2.1.1 Sistem

Menurut Kristanto (2018:3) “Sistem dapat diartikan sebagai sekumpulan komponen (elemen atau unsur) yang saling berkaitan hingga (dapat) mempengaruhi (sesuai dengan fungsi masing-masing) satu sama lainnya dengan urutan/prosedur tertentu dalam mencapai suatu tujuan atau menghasilkan suatu output”.

Pengertian sistem menurut Meilinda (2018), Sistem merupakan kumpulan himpunan atau komponen dari unsur yang saling bergantung dan berinteraksi untuk sebuah tujuan yang bersifat sama.

Berdasarkan pengertian di atas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2.1.2 Informasi

Pengertian menurut Romney dan Steinbart (2015:4), Informasi adalah data yang telah di kelola dan di proses pengambilan keputusan. Sebagaimana perannya, penggunaan pembuatan keputusan yang lebih baik sebagaimana kuantitas dan kualitas dari peningkatan informasi.

Pengertian menurut Sutabri (2016) dalam buku Sistem Informasi Manajemen, Informasi adalah data yang telah diklasifikasi atau diolah atau interpretasi untuk digunakan dalam proses pengambilan keputusan.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa konsep sistem informasi adalah data yang diolah untuk mengambil keputusan yang berguna bagi penggunanya.

2.1.3 Sistem Informasi

Menurut Muslihudin dan Oktafianto (2016:10) “Sistem informasi adalah hubungan antara data dan informasi, dan merupakan entitas penting yang merupakan sistem informasi”.

Menurut Anggraeni dan Irviani (2017:2) “Sistem informasi adalah kombinasi terorganisir dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang digunakan untuk mengumpulkan, mengubah, dan menyebarkan informasi dalam suatu organisasi”.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa sistem informasi adalah suatu kegiatan atau kegiatan dari suatu sistem informasi yang dibentuk oleh serangkaian proses yang berupa komponen-komponen.

2.2 Metode Waterfall

Menurut Rosa dan Shalahuddin (2018:28), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (support)”. Adapun metode air terjun menurut Rosa dan Shalahuddin (2018:29) yaitu:

1. Analisis

Kebutuhan Perangkat Lunak Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi

kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi-langkah yang berfokus pada desain program perangkat lunak, termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan proses pengkodean. Fase ini mengubah kebutuhan perangkat lunak dari fase analisis kebutuhan menjadi

representasi desain untuk implementasi ke dalam program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

Desain harus diterjemahkan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer berdasarkan desain yang dibuat selama tahap desain.

4. Pengujian

Pengujian berfokus pada logika dan fungsionalitas perangkat lunak dan memastikan bahwa semua bagian diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan output yang dihasilkan sesuai dengan yang diharapkan.

5. Pendukung (support) atau Pemeliharaan (maintenance)

Perangkat lunak dapat diubah ketika dikirim ke pengguna. Perubahan dapat terjadi karena bug yang tidak terdeteksi yang terjadi selama pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Fase dukungan atau pemeliharaan dapat mengulangi proses pengembangan dari analisis spesifikasi hingga mengubah perangkat lunak yang ada, akan tetapi tidak mengulangi pembuatan perangkat lunak (*software*) baru.

2.3 Monitoring

Menurut Peraturan Pemerintah Nomor 39 Tahun 2006, pengertian *monitoring* adalah suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan dan tindakan selanjutnya yang diperlukan. Tindakan tersebut diperlukan jika hasil pengamatan menunjukkan adanya hal atau kondisi yang tidak sesuai dengan yang diinginkan. *Monitoring* juga berfungsi untuk mengamati atau mengetahui perkembangan dan kemajuan, identifikasi dan permasalahan, antisipasinya dan solusi pemecahannya (“Nomor, P. P. (39). Tahun 2006 tentang Tata Cara Pengendalian dan Evaluasi Pelaksanaan Rencana Pembangunan.,” t.t.).

2.4 Bantuan Dana Desa

Menurut Buku Saku Dana Desa yang dikeluarkan oleh Kementerian Keuangan (2017) Pengertian Dana Desa adalah dana Anggaran Pendapatan Belanja Negara yang digunakan untuk Desa yang dikirim untuk pelaksanaan pembangunan dan pemberdayaan masyarakat melalui Anggaran Pendapatan Belanja Daerah kabupaten/kota.

2.4.1 Pembangunan Desa

Pembangunan desa merupakan upaya peningkatan kualitas hidup dan kualitas hidup untuk sebesar-besarnya kesejahteraan masyarakat desa. Pembangunan desa tidak dapat dilepaskan dari konteks pengelolaan pembangunan daerah di tingkat kabupaten dan provinsi, karena status desa dalam konteks yang lebih luas (sosial, ekonomi, akses pasar dan politik) harus antar partisi dan antar partisi. Pembangunan pedesaan memainkan peran penting dalam proyek-proyek pembangunan nasional. Karena desa ini memiliki berbagai macam pembangunan, karena merupakan tumpuan pembangunan. Pembangunan desa bertujuan untuk meningkatkan kualitas hidup dan kehidupan masyarakat desa.

Ada banyak hal yang harus dilakukan dalam pembangunan pedesaan. Pelaksanaan pembangunan desa harus mengacu pada pencapaian tujuan pembangunan, yaitu mewujudkan kehidupan masyarakat desa yang mandiri, maju, sejahtera, dan berkeadilan. Karena pembangunan desa ini merupakan salah satu agenda besar untuk mengawal implementasi UU No. 6 Tahun 2014 tentang Desa yang dilaksanakan secara sistematis, konsisten, dan berkelanjutan dengan jalan fasilitasi, supervisi, dan pendampingan (Buku Bantu Pengelolaan Pembangunan Desa Berdasarkan Undang-Undang Nomor 6 Tahun 2014 tentang Desa t.t.).

2.4.2 Tujuan Pembangunan Desa

Dalam sebuah pembangunan desa, maka akan terlaksana dengan baik dan terarah sesuai dengan tujuan awal. Secara khusus dari pembangunan desa sebagai berikut:

1. Meningkatkan kemampuan kelembagaan masyarakat di tingkat desa dalam penyusunan perencanaan pembangunan secara partisipatif;

2. Meningkatkan keterlibatan seluruh elemen masyarakat dalam memberikan makna dalam perencanaan pembangunan;
3. Meningkatkan transparansi dan akuntabilitas pembangunan; dan
4. Menghasilkan keterpaduan antar bidang/sector dan kelembagaan dalam kerangka.

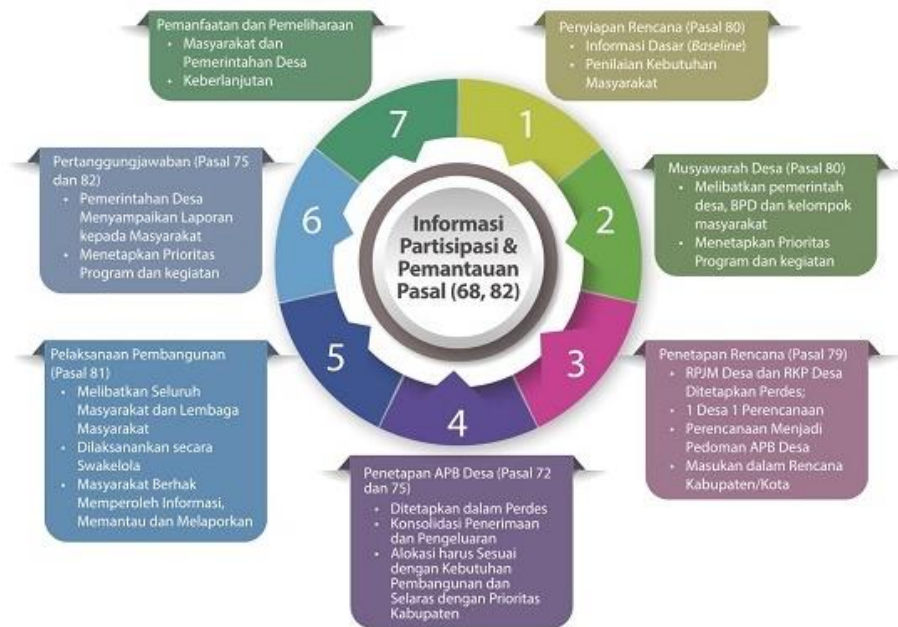
Memaknai beberapa pendapat mengenai tujuan pembangunan desa menurut UU Desa, adalah meningkatkan kesejahteraan hidup manusia serta penanggulangan kemiskinan melalui penyediaan pemenuhan kebutuhan dasar, pembangunan sarana dan prasarana desa, pengembangan potensi ekonomi lokal dan pemanfaatan sumber daya alam dan lingkungan secara berkelanjutan.

Pembangunan desa dilaksanakan dengan mengedepankan semangat kebersamaan, kekeluargaan, dan kegotongroyongan guna mewujudkan pengarusutamaan perdamaian serta keadilan sosial (Buku Bantu Pengelolaan Pembangunan Desa Berdasarkan Undang-Undang Nomor 6 Tahun 2014 tentang Desa t.t.).

2.4.3 Siklus Pembangunan Desa

Alur atau proses pembangunan desa dalam rentang waktu satu tahun (Januari–Desember) yang digambarkan sebagai berikut:

SIKLUS PEMBANGUNAN DESA



Gambar 2.1. Siklus Pembangunan Desa

Sumber : *Buku Bantu Pengelolaan Pembangunan Desa Berdasarkan Undang-Undang No*

2.5 Unified Modeling Language

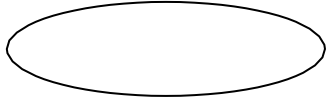
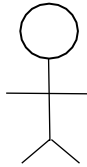

Menurut (Rosa dan Shalahuddin 2018), “*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan dalam industri untuk mendefinisikan persyaratan, melakukan analisis dan desain, dan menggambarkan arsitektur dalam pemrograman berorientasi objek. UML adalah bahasa visualisasi untuk pemodelan dan sistem komunikasi menggunakan diagram dan teks pendukung. UML hanya untuk pemodelan.

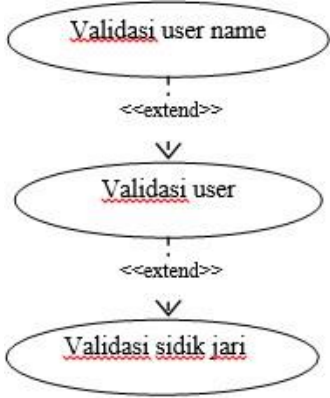

2.5.1 Jenis-jenis Unified Modeling language

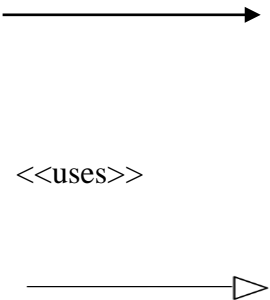
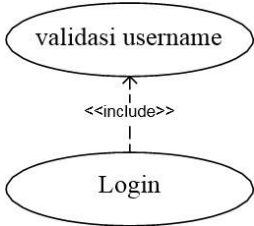
1. Use Case Model

Sukamto dan Shalahuddin (2018:155), “*Use case* atau diagram *use case* adalah suatu pemodelan perilaku (*behavior*) dari suatu sistem informasi yang akan dibuat. Sebuah *use case* menggambarkan interaksi antara satu atau lebih aktor dan sistem informasi yang akan dibuat. Secara dasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sistem informasi dan siapa yang memiliki akses ke fungsi tersebut”. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1. Simbol-simbol Use Case Diagram

| No. | Simbol | Deskripsi |
|-----|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <p><i>Use case</i></p>  <p>Nama use case</p> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> . |
| 2. | <p>Aktor/<i>actor</i></p>  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 3. | <p>Assosiasi/<i>association</i></p>  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| 4. | <p>Exstensi/<i>extend</i></p> | Relasi antara <i>use case</i> tambahan dengan <i>use case</i> , bahkan jika tidak |

| | | |
|----|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p><<extend>></p> | <p>ada <i>use case</i> tambahan, <i>use case</i> yang ditambahkan dapat eksis secara independen, mirip dengan prinsip pewarisan dalam pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama yang sama dengan yang ditambahkan kasus penggunaan, misalnya</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p> |
| 5. | <p>Generalisasi/<i>generalization</i></p>  | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya : arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya(umum)</p> |
| 6. | <p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><<include>></p> | <p>Relasi antara <i>use case</i> tambahan ke <i>use case</i> di mana <i>use case</i> yang ditambahkan membutuhkan <i>use</i></p> |



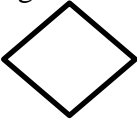


| | | |
|--|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| |  | <p><i>case</i> ini untuk menjalankan fungsinya atau sebagai kondisi eksekusinya.:</p> <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :  <pre> graph BT Login(Login) -.-> <<include>> validasi_username(validasi username) </pre> <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut : Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. |
|--|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

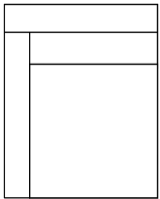
Sumber : Sukamto dan Shalahuddin (2018:156-158)

2. Activity Diagram

Sukamto dan Shalahuddin (2018:161), “*Activity* diagram atau diagram aktivitas menggambarkan alur kerja (*workflow*) atau aktivitas dari suatu sistem atau proses bisnis atau menu dalam perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem daripada apa yang dilakukan aktor, sehingga sistem dapat melakukan aktivitas”. Berikut adalah simbol- simbol yang ada pada diagram aktivitas

Tabel 2.2. Simbol-simbol Activity Diagram

| No. | Simbol | Deskripsi |
|-----|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| 1. | Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2. | Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3. | Percabangan/ <i>decision</i>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4. | Penggabungan/ <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5. | Status akhir  | Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

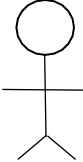

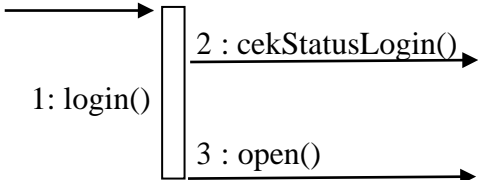
| | | |
|----|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 6. | Swimlane  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
|----|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|

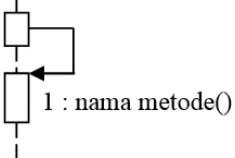
Sumber : Rosa dan Shalahuddin (2018:162-163)

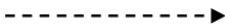
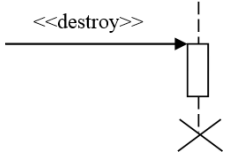
3. Sequence Diagram

Sukanto dan Shalahuddin (2018:165), “Diagram sekuen menggambarkan perilaku objek dalam kasus penggunaan dengan menggambarkan siklus hidup mereka dan pesan yang dikirim dan diterima di antara mereka. Oleh karena itu, untuk menggambarkan diagram urutan, perlu dipahami objek yang terlibat dalam *use case* dan metode yang dimiliki oleh kelas yang diinstansiasi ke objek tersebut. Sequence diagram juga perlu dibuat untuk melihat skenario yang ada pada *use case*. Jumlah sequence diagram yang harus digambar setidaknya sebanyak *use case* yang mendefinisikan prosesnya sendiri, atau yang terpenting semua *use case* yang mendefinisikan interaksi pesan dimasukkan ke dalam sequence diagram, sehingga semakin banyak *use case* yang didefinisikan maka semakin banyak sequence. diagram harus dibuat lebih banyak.”. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.3. Simbol-simbol Sequence Diagram

| No. | Simbol | Deskripsi |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <p data-bbox="432 383 512 409">Aktor</p>  <p data-bbox="432 656 499 683">Atau</p> <p data-bbox="568 763 719 790"><u>Nama aktor</u></p> <p data-bbox="432 875 671 902">Tanpa waktu aktif</p> | <p data-bbox="890 383 1358 853">Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi akan dibuat di luar sistem informasi yang akan dibuat, sehingga walaupun simbol aktor adalah gambaran seseorang, aktor tersebut belum tentu orang, biasanya diwakili oleh penggunaan kata benda di awal frase nama aktor.</p> |
| 2. | <p data-bbox="432 934 687 960">Garis hidup/<i>lifeline</i></p>  | <p data-bbox="890 934 1353 960">Menyatakan kehidupan suatu objek</p> |
| 3. | <p data-bbox="432 1211 512 1238">Objek</p> <p data-bbox="432 1373 759 1400"><u>Nama objek : nama kelas</u></p> | <p data-bbox="890 1211 1358 1294">Menyatakan objek yang berinteraksipesan</p> |
| 4. | <p data-bbox="432 1487 592 1514">Waktu aktif</p> | <p data-bbox="890 1487 1358 1738">Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukandi dalamnya, misalnya</p>  |

| | | |
|----|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Maka cek StatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif |
| 5. | Pesan tipe <i>create</i> <<create>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6. | Pesan tipe <i>call</i> 1 : nama_metode() → → | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi |
| 7. | Pesan tipe <i>send</i> 1 : masukkan → | Menyatakan bahwa suatu objek mengirimkan data masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 8. | Pesan tipe <i>return</i> 1 : keluaran | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek |

| | | |
|----|--------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| |  | yang menerima kembalian. |
| 9. | Pesan tipe <i>destroy</i>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i> |

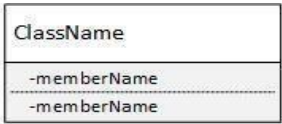

Sumber: Sukamto dan Shalahuddin (2018:165-167)



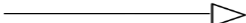
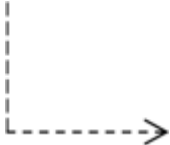

4. Class Diagram

Sukamto dan Shalahuddin (2018:141), “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.4. Simbol-simbol Class Diagram

| No. | Simbol | Deskripsi |
|-----|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| 1. | Kelas  | Kelas pada struktur system |
| 2. | Antarmuka/ <i>interface</i>  | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |

| | | |
|----|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 3. | Asosiasi/ <i>association</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| 4. | Asosiasi berarah/ <i>directed association</i>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| 5. | Generalisasi  | Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus) |
| 6. | Kebergantungan/ <i>dependensi</i>  | Relasi antar kelas dengan makna kebergantungan antar kelas |
| 7. | Agregasi/ <i>aggregation</i>  | Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>) |

Sumber : Sukamto dan Shalahuddin (2018:146-147)

5. Kamus Data

Sukamto, Ariani Rosa, dan M. Shalahuddin (2018), mengemukakan bahwa kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada *DFD*. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) data keluaran (*output*) dapat dipahami secara umum.

2.5.2 HyperText Preprocessor (PHP)

Menurut Sidik (2017:4), "PHP secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di

server web, dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*."

Menurut Harianto,dkk (2019:13), "PHP merupakan software Open-Source yang disebarakan dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya".

Berdasarkan pengertian di atas, bahwa dapat disimpulkan PHP adalah bahasa pemrograman berupa data yang dikelola melalui sebuah sistem untuk input,output untuk membuat website.

2.5.3 XAMPP

Menurut Aryanto dalam, Kesuma dan Juniati (2020), XAMPP merupakan aplikasi perangkat lunak untuk pemrograman dan database yang didalamnya terdapat Apache HTTP Server, database MySQL dan bahasa pemrograman PHP.

2.5.4 Database

Menurut Setiawan (2017), Basis data atau database adalah kumpulan tabel yang berisi sejumlah besar data, serta kumpulan baris (field) dan kolom (columns). Disini kita akan menyimpan data dari project yang kita buat.

Pamungkas (2017) basis Data ialah suatu kumpulan data yang saling terhubung dan disimpan secara bersamaan pada sebuah media, yang diorganisasikan berdasarkan suatu skema atau struktur tertentu dengan software agar bias digunakan untuk manipulasi kegunaan tertentu.

2.5.5 MySQL (My Structure Query Language)

Menurut Harianto,dkk (2019:13), "MySQL adalah salah satu jenis data-base server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai sumber dan pengelolaan datanya".

2.5.6 HTML

Sulistiono (2018), HTML (Hypertext Markup Language) adalah bahasa markup yang digunakan untuk membuat suatu halaman website, menampilkan berbagai macam informasi seperti gambar, teks, suara, dan video yang ada pada web internet,

ditulis dalam sebuah berkas format ASCII supaya dapat menghasilkan sebuah tampilan wujud yang terintegrasi.

2.5.7 CSS (Cascading Style Sheet)

Menurut Andi dalam (Apriyanto & Ramadhan, 2017) menjelaskan bahwa “CSS adalah suatu kumpulan kode-kode untuk memformat atau mengendalikan tampilan isi dalam suatu halaman web”.

2.5.8 JavaScript

Sulistiono (2018), JavaScript merupakan suatu bahasa yang digunakan untuk membuat sebuah program yang agar dokumen HTML yang ditampilkan dalam browser agar menjadi lebih interaktif dan tidak sekedar indah saja. JavaScript juga memberikan beberapa fungsionalitas ke suatu halaman web.

2.5.9 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks *editor* yang ringan dan handal dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*. Teks *editor* ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang *via marketplace* Visual Studio Code (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dst). (Permana & Romadhon, 2019).

2.5.10 Hasil Penelitian Sebelumnya

| No. | Data Jurnal / Makalah | Keterangan |
|-----|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Judul | Sistem Informasi Monitoring Kinerja Sdm |
| 2 | Lokasi dan Subjek Penelitian | Pt Pln Unit Pelaksana Pembangkitan Tarahan |
| 3 | Masalah | HCR adalah kesiapan kompetensi dan kompetensi sumber daya manusia untuk mendukung implementasi strategi perusahaan, sedangkan OCR adalah kesiapan organisasi untuk mengintegrasikan dan mengkoordinasikan faktor-faktor yang mendorong perbaikan dan pembelajaran berkelanjutan untuk mendukung agenda perubahan organisasi melalui implementasi strategi perusahaan. HCR dan OCR sendiri didorong oleh manusia. Departemen sumber daya merupakan faktor terpenting dalam mengevaluasi nilai kinerja organisasi, terutama penilaian kinerja. Kesalahan kecil dapat membuat perbedaan besar jika nilai kinerja departemen SDM gagal. |
| 4 | Kesimpulan | Merancang dan mengimplementasikan sistem monitoring kinerja dengan menggunakan metode Key Performance Indicator (KPI) dapat membantu bagian SDM dalam memantau kinerja perusahaan untuk membantu mencapai Nilai Kinerja Organisasi (NKO) dan memudahkan perusahaan dalam memonitoring target – target kinerja. |
| 5 | Metode | Prototype |

| No. | Data Jurnal / Makalah | Keterangan |
|-----|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Judul | Pembangunan Sistem Informasi Monitoring Pengelolaan Kegiatan Karang Taruna Kecamatan Padalarang |
| 2 | Lokasi dan Subjek Penelitian | Karang Taruna Kecamatan Padalarang |
| 3 | Masalah | Karang Taruna Kabupaten Padalarang belum ada informasi pelaksanaan kegiatan yang dilakukan Karang Taruna Desa. Akibatnya, efek pemantauan secara keseluruhan terhadap kegiatan organisasi pemuda jalanan tidak baik. Minimnya pengawasan organisasi dan unit pemuda desa mengakibatkan pemerintah kecamatan tidak mampu menggambarkan capaian atau tingkat perkembangannya. Kegiatan yang dilakukan oleh organisasi pemuda desa. |
| 4 | Kesimpulan | Membangun sistem informasi untuk memantau perkembangan organisasi kepemudaan di setiap unit dan desa di Kecamatan Padalarang, dan membangun sistem yang dapat memantau kegiatan dan menghasilkan persentase perkembangan organisasi kepemudaan di setiap desa. |
| 5 | Metode | Metode Waterfall |

(R et al., 2019) Meryana Putri R*, Faiza Renaldi, Irma Santikarama Jurusan Informatika Fakultas Sains dan Informatika Universitas Jenderal Achmad Yani (*Seminar Nasional Aplikasi Teknologi Informasi (SNATi) 2019 ISSN: 1907-5022 Yogyakarta, 3 Agustus 2019*)

| No. | Data Jurnal / Makalah | Keterangan |
|-----|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Judul | Sistem Informasi Monitoring Dan Evaluasi Pelaksanaan Anggaran Pendapatan Dan Belanja Daerah (Apbd) |
| 2 | Lokasi dan Subjek Penelitian | Kantor Dinas Pendidikan Kabupaten Soppeng |
| 3 | Masalah | Laporan pelaksanaan rencana/kegiatan, progres pelaksanaan setiap rencana/kegiatan setiap bulan, sampai dengan pelaksanaan hasil yang akan dicapai rencana/kegiatan dan evaluasinya, yang dibagi ke dalam beberapa folder di komputer. mengelola sistem memakan waktu Lebih lama, data mungkin hilang, tidak valid, dan tidak efisien. |
| 4 | Kesimpulan | Dengan adanya sebuah Sistem Monitoring dan Evaluasi Anggaran Pendapatan dan Belanja Daerah (APBD) pada Dinas Pendidikan Soppeng dapat memberikan informasi yang sesuai dengan yang diinginkan dan dapat membantu Pemerintah dalam memonitoring dan mengevaluasi anggaran yang sedang berjalan maupun yang telah terealisasi. |
| 5 | Metode | <ul style="list-style-type: none"> - Observasi - Wawancara - Studi literatur - menggunakan pemrograman visual basic 6.0 |

| No. | Data Jurnal / Makalah | Keterangan |
|-----|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Judul | Penerapan Metode <i>Agile</i> Untuk Membangun Sistem Informasi Monitoring Santri |
| 2 | Lokasi dan Subjek Penelitian | Pondok Modern Asy-Syifa Balikpapan |
| 3 | Masalah | Jumlah santri semakin bertambah dan perlu adanya pencatatan data nilai, pelanggaran dan perijinan yang tidak lagi dilakukan secara manual di buku, dan Pondok Modern Asy-Syifa Balikpapan perlu memantau nilai, pelanggaran dan izin sebagai bahan pertimbangan dalam pemberian nilai siswa , menghukum siswa, atau memberikan izin kepada santri. |
| 4 | Kesimpulan | Diperoleh kesimpulan berupa sistem informasi yang dapat melakukan pencatatan prestasi, pelanggaran dan perizinan santri, serta dapat melakukan monitoring prestasi, pelanggaran dan perizinan santri yang berbasis <i>website</i> sesuai dengan kebutuhan Pondok Modern Asy-Syifa Balikpapan, yaitu Sistem Informasi Monitoring Santri (SIMS) Pondok Modern Asy-Syifa Balikpapan berhasil dikembangkan dengan diselesaikannya 6 <i>sprint</i> , 36 <i>product backlog</i> yang memiliki total 72 <i>story points</i> . |
| 5 | Metode | Metode <i>agile</i> dengan <i>model scrum</i> untuk pengembangan Sistem Informasi Monitoring Santri (SIMS). |

(Buana et al., 2021) Sayid Esa Tri Buana, Lovinta Happy Atrinawati, M. Gilvy Langgawan Putra Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK) ke 5 Tahun 2021