# BAB II TINJAUAN PUSTAKA

## 2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh (Cahyadi & Susanto, 2020) dengan judul “Rancang Bangun Sistem Informasi Bimbingan Konseling Di SMK Tamansiswa Mojokerto Berbasis Web Menggunakan Model *Waterfall*”. Penelitian ini bertujuan untuk membuat sistem bimbingan konseling dengan menggunakan model *waterfall*. Penelitian saat ini dilakukan pengembangan pada penyampaian informasi pelanggaran kepada wali siswa melalui kontak WhatsApp.

Penelitian yang dilakukan oleh (Hanafi dkk., 2020) dengan judul “Sistem Informasi Bimbingan Konseling Berbasis *Web* (Studi Kasus: Smk Negeri 1 Mojoagung)”. Penelitian ini bertujuan untuk merancang sistem informasi bimbingan konseling berbasis web yang akan memudahkan pencatatan pelanggaran bagi guru BK. Kelebihan pada Penelitian saat ini yaitu dilakukan mengembangan dalam sektor memperketat pelanggaran dengan cara pengiriman Informasi berupa dokumen surat peringatan yang bisa di tujukan ke wali siswa.

Refrensi Penelitian terakhir dilakukan oleh (Baijuri dkk., 2022) dengan judul “Sistem Informasi Pengolahan Data Pelanggaran Siswi SMK Ibrahimy 1 Berbasis Web”. Penelitian ini bertujuan untuk memudahkan pencatatan bimbingan dan konseling yang masih menggunakan *Microsoft* *excel* dalam membuat data laporan perbulan maupun input data pelanggaran siswa, selain itu masih mengalami kesulitan dalam mencari data skor siswa. Kelebihan dari Penelitian saat ini dilakukan pengembangan dengan menambahkan peran guru bk sebagai sarana bimbingan konseling kepada siswa.Tabel 2. 1 Tabel Jurnal perbandingan dengan penelitian sebelumnya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Judul** | **Peneliti** | **Kesimpulan** | **Kelebihan Penelitian Saat ini** | **Kekurangan**  **Penelitian saat ini** |
| SISTEM INFORMASI BIMBINGAN KONSELING BERBASIS  WEB (STUDI KASUS : SMK NEGERI 1 MOJOAGUNG) | Rizwan Hanafi), Nur Cahyo Wibowo) ,Agung Brastama Putra) | Kesimpulan dari penelitian ini adalah sistem dapat memproses pelanggaran , memasukan data prestasi dan artikel. | Penelitian saat ini dilakukan mengembangan dalam sektor memperketat pelanggaran dengan cara pengiriman Informasi berupa dokumen surat peringatan yang bisa di tujukan ke wali siswa. | Penelitian saat ini tidak mencantumkan data prestasi siswa. |
| SISTEM INFORMASI PENGOLAHAN DATA PELANGGARAN SISWI  SMK IBRAHIMY 1 BERBASIS WEB | Achmad Baijuri1  , Fasiha1  ,Ahmad Musaddad2 | Memberikan kemudahan bagi guru BK. Oleh karena itu, sistem informasi ini dirancang dengan tujuan agar dapat digunakan dan beroperasi secara optimal oleh pengguna. | Penelitian saat ini dilakukan pengembangan dengan menambahkan peran guru bk sebagai sarana bimbingan konseling kepada siswa. | Laporan rekap jumlah bimbingan konseling siswa yang di tujukan ke bk tidak ada. |
| RANCANG BANGUN SISTEM INFORMASI BIMBINGAN  KONSELING DI SMK TAMANSISWA MOJOKERTO BERBASIS  WEB MENGGUNAKAN MODEL WATERFALL | Cornelia Miyastuti Cahyadi1  ,Gaguk Susanto2 | Hasil pengujian sistem informasi menggunakan UAT yang  memperoleh nilai 74,25 pada aspek sistem, 75,5 pada aspek pengguna dan 75,75 pada  aspek interaksi dengan skala nilai tertinggi yaitu 80. | Penelitian saat ini dilakukan pengembangan pada penyampaian informasi pelanggaran kepada wali siswa melalui kontak WhatsApp. | Tidak tersedianya menu kunjungan rumah. |

## 2.2 Teori Terkait

### 2.2.1 Sistem Informasi

Sistem informasi merupakan gabungan dari pengertian sistem dan informasi sebagaimana uraian di atas (Abdurahman, 2018). Sistem informasi adalah kegiatan dari prosedur yang di organisasikan yang digunakan untuk menyediakan informasi pengambilan keputusan dan pengendalian pada sebuah organisasi (Kurniawan dkk., 2020).

### 2.2.2 *Flowchart*

*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Abdurahman, 2018).

Berikut akan dijelaskan mengenai simbol-simbol *flowchart* yang dibagi kedalam 3 kategori (Rosaly dkk., 2019), diantaranya:

1. Simbol Arus (*Flow Direction Symbols*)

Biasanya simbol yang termasuk kedalamketegori ini digunakan sebagai simbol penghubung. Beberapa simbol yang termasuk ke dalam kategori ini, yaitu:

Tabel 2. Simbol Arus Flowchart

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Flow Direction Symbol / Connecting Line* | Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses |
|  | *Communication Link* | Berfungsi untuk transmisi data dari satu lokasi ke lokasi lain |
|  | *Connector* | Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama |
|  | *Offline Connector* | Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda |

1. Simbol Proses (*Processing Symbols*)

Sesuai dengan namanya, simbol proses digunakan untuk menyatakan simbol yang berkaitan dengan serangkaian proses yang dilakukan. Berikut beberapa simbol yang termasuk kedalam bagian proses, yaitu:

Tabel 2. Simbol Proses *Flowchart*

|  |  |  |
| --- | --- | --- |
| Bentuk | Nama | Kegunaan |
|  | *Processing* | Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer |
|  | *Decision* | Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu |
|  | *Predefined Process* | Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberikan harga awal |
|  | *Terminal /Terminator* | Digunakan untuk memulai atau mengakhiri program |
|  | *Manual Input* | Digunakan untuk menginputkan data secara manual dengan keyboard |

1. Simbol I/O (*Input*-*Output*)

Simbol yang termasuk kedalam bagian input-output berkaitan dengan masukan dan keluaran. Berikut beberapa simbol yang termasuk, yaitu:

Tabel 2. Simbol *Input*/*Output*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Input/Output* | Digunakan untuk menyatakan input dan output tanpa melihat jenisnya |
|  | *Document* | Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen |
|  | *Display* | Digunakan untuk menyatakan keluaran melalui layar monitor |

### 2.2.3 *Unified Modeling Language* (UML)

UML merupakan sebuah standar Bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman *object oriented* (Kurniawan dkk., 2020)*.*

UML (*Unified Modeling Langguage*) adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi. UML menyediakan model- model yang tepat, tidak ambigu, dan lengkap. Secara khusus UML menspesifikasi langkah-langkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak (Nugroho & Rohimi, 2020).

### 2.2.4 *Use Case Diagram*

*Use case* diagram adalah gambaran grafis dari beberapa atau semua *actor, use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case* diagram tidak menjelaskan secara detil tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat (Kurniawan dkk., 2020).

*Use Case* pada dasarnya merupakan gambaran dari proses sistem secara keseluruhan yang melibatkan actor dalam hal pengguna. *Use case* adalah cara untuk menunjukan *stakeholder* sistem akan berinteraksi dengan sistem. Mengebangakan *use case* membantu memahami persyaratan sistem secara *detail* (Nugroho & Rohimi, 2020).

Berikut adalah simbol *use case* diagram (Ismail dkk., 2021).

Tabel 2. Simbol *Use Case* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran  pengguna mainkan ketika berinteraksi dengan *use case.* |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi  elemen yang bergantung padanya  elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan  struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang  diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang  menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang  ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor. |
| 9 |  | *Collaborati on* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

### 2.2.5 Activity Diagram

*Activity diagram* adalah pemodelan yang dilakukan pada suatu sistem dan menggambarkan aktivitas sistem berjalan. *Activity diagram* di gunakan sebagai penjelelasan aktivitas program tanpa melihat koding atau tampilan (Kurniawan dkk., 2020).

Berikut adalah simbol *activity* *case* diagram (Ismail et al., 2021).

Tabel 2. Simbol *Activity* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Activity* | Memperlihatkan bagaimana masing- masing kelas antarmuka saling berinteraksi satu sama lain. |
| 2 |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi. |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 |  | *Actifity Final Node* | Bagaimana objek di bentuk dan di hancurkan. |
| 5 |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. |

### 2.2.6 Class Diagram

*Class diagram* adalah diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode), dan hubungan antar kelas (Nugroho & Rohimi, 2020).

Berikut adalah simbol *class* *case* diagram (Ismail dkk., 2021)

Tabel 2. Simbol *Class* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Generalization* | Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor). |
| 2 |  | *Nary*  *Association* | Upaya untuk menghindari  asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu  aktor. |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempegaruhi elemen yang bergantung padanya elemen  yang tidak mandiri. |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek  lainnya. |

### 2.2.7 Pelanggaran

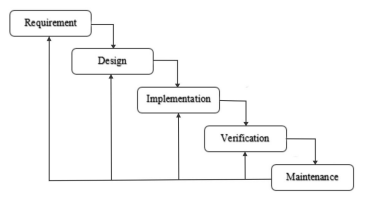
Pelanggaran tata tertib adalah sikap tidak perhatian, misalnya sering meninggalkan kelas, bercakap-cakap selama pelajaran berlangsung, tidak menjawab pertanyaan guru, tidak mengerjakan tugas, lambat mengerjakan tugas (Mulyanto & Ristina, 2018).

### 2.2.8 Tata Tertib

Tata tertib sekolah ialah ketentuan-ketentuan yang mengatur kehidupan sekolah sehari-hari dan mengandung sanksi terhadap pelanggarannya (Mulyanto & Ristina, 2018).

### 2.2.9 Metode Waterfall

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam Software Engineering (SE). saat ini model waterfall merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya. (Aceng, 2020).



Gambar 2. Metode *Waterfall*