

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang diambil merupakan penelitian yang sudah ada terkait sebagai bahan pendukung penelitian yang akan dilakukan. Tinjauan pustaka ini merupakan beberapa rancangan dan metode yang akan menjadi bahan referensi dalam perancangan *website*, penulis telah mengumpulkan beberapa tinjauan pustaka yang dapat dilihat pada Tabel 2.1 dibawah ini:

Tabel 2.1 Tinjauan Literatur

Nomor Literatur	Penulis	Tahun	Judul
Literatur 1	Ela Nurelasari	2020	Perancangan Sistem Informasi Akademik Pada Sekolah Menengah Pertama Berbasis Web
Literatur 2	Reza Sangga Rasefta & Shinta Esabella	2020	Sistem Informasi Akademik SMK Negeri 3 Sumbawa Besar Berbasis Web
Literatur 3	Ahmad Yani, Ahmad Syauki, Siti Marlina	2019	Rancang Bangun Sistem Informasi Akademik Berbasis Web Pada Madrasah Aliyah Attaqwa Tangerang.

Tabel 2.2 Tinjauan Literatur (Lanjutan)

Nomor Literatur	Penulis	Tahun	Judul
Literatur 4	Syamsul Bakhri	2018	Sistem Informasi Akademik Mahasiswa Berbasis Web Pada Akademi Kebidanan Yaspem Tugu Ibu Kota Jakarta
Literatur 5	Khaerul Anam & Asep Taufik Muharram	2018	Analisis dan Perancangan Sistem Informasi Akademik Berbasis Web Pada MI ALMusyidiyyah AL-‘Asyiroutussyafi’iyyah
Literatur 6	Tubagus Riko Rivanthio	2017	Perancangan Sistem Informasi Akademik Berbasis <i>Website</i> Pada Sekolah Tinggi Analisis Bakti Asih Bandung.

1.1.1 Tinjauan Terhadap Literatur 1

Penelitian oleh Ela Nurelasari (2020) adalah sebuah sistem informasi akademik berbasis *web* yang dapat digunakan oleh pihak sekolah untuk pengelolaan data agar dapat meminimalisir penggunaan kertas karena pendataan yang dilakukan administrator secara sistem komputerisasi.

1.1.2 Tinjauan Terhadap Literatur 2

Penelitian oleh Reza Sangga Rasefta & Shinta Esabella (2020) berupa *website* yang menggunakan metode *Extreme Programming* yang dimana telah menghasilkan sistem dengan proses pembuatan yang mudah.

1.1.3 Tinjauan Terhadap Literatur 3

Penelitian oleh Ahmad Yani, Ahmad Syauki, Siti Marlina (2020) adalah sebuah sistem informasi akademik berbasis *web* yang bertujuan untuk memudahkan para siswa/i dan guru untuk tidak perlu bertemu secara langsung untuk berinteraksi ataupun untuk mengakses informasi seputar kegiatan akademik.

1.1.4 Tinjauan Terhadap Literatur 4

Penelitian oleh Ahmad Yani, Ahmad Syauki, Siti Marlina (2020) adalah sebuah sistem informasi akademik berbasis *web* yang bertujuan untuk memudahkan para siswa/i dan guru untuk tidak perlu bertemu secara langsung untuk berinteraksi ataupun untuk mengakses informasi seputar kegiatan akademik.

1.1.5 Tinjauan Terhadap Literatur 5

Penelitian oleh Khaerul Anam & Asep Taufik Muharram (2018) adalah peneliti menyebutkan bahasa informasi akademik masih bersifat *paper base* atau menggunakan kertas secara manual dikarenakan cara ini sangat rentan dari berbagai macam kesalahan seperti kesalahan teknis *error (technical error)*, ataupun kesalahan manusia (*human error*). Karena itu dibuatlah sistem informasi akademik.

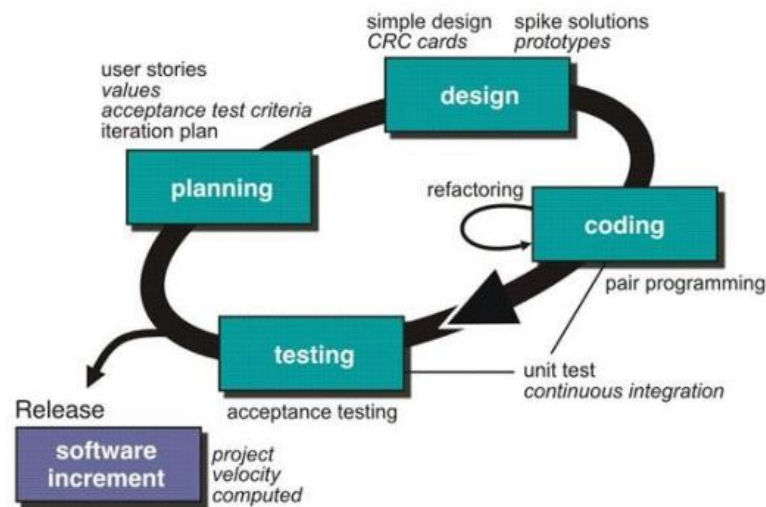
1.1.6 Tinjauan Terhadap Literatur 6

Penelitian oleh Tubagus Riko Rivanthio (2017) adalah sebuah rancangan sistem berbasis *web* yang memudahkan dalam pengelolaan data dan pelayanan

BAAK terhadap dosen ataupun mahasiswa yang dapat dilakukan secara efektif dan efisien.

2.2 Metode Extreme Programming

Extreme Programming adalah metodologi pengembangan perangkat lunak yang ditunjukkan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pro pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi (Novitasari et al., 2021). Tahapan pengembangan perangkat lunak dengan XP meliputi: *planning* (perencanaan), *design* (perancangan), *coding* (pengkodean), dan *testing* (pengujian) (Borman et al., 2020).



Gambar 2.1 Tahapan *Extreme Programming* (Borman et al., 2020)

1. *Planning* (Perencanaan)

Tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem, dengan memahami konteks dari sistem yang akan dibuat dan tahapan ini menentukan alur sistem sesuai kebutuhan pengguna ataupun pengembang.

2. *Design* (Perancangan)

Tahapan ini pengembang melakukan perancangan dengan membuat sebuah pemodelan, yang dimulai dari pemodelan sistem, kemudian pemodelan arsitektur dan pemodelan basis data untuk memberikan gambaran sistem yang akan dibangun. Tahapan ini dapat menggunakan CRC (*Class Responsibility Collaborator*). CRC berfungsi untuk memetakan kelas-kelas yang akan dibangun. Selanjutnya pengembang membuat *Prototype* yang terdiri dari *user interface* untuk memudahkan klien melihat sistem. *Spike Solution* dilakukan oleh pengguna agar mendapatkan kesesuaian terhadap klien dan pengembang.

3. *Coding* (Pengkodean)

Tahapan ini pengembang menyesuaikan terhadap story klien sehingga sistem yang dibangun sesuai kebutuhan, terdapat 2 (Dua) proses pengkodean yaitu:

- a. *Pair Programming* adalah tahap sistem yang dibangun dengan bahasa pemrogramman dan media penyimpanan yang telah disepakati.
- b. *Refactory* adalah tahapan yang dilakukan ketika terjadi ketidak sesuaian dalam pengkodean sehingga dilakukan perbaikan untuk mendapatkan hasil yang sesuai.

4. *Testing* (Pengujian)

Tahap pengujian dilakukan oleh klien sebagai *user* dengan melakukan uji coba kesesuaian terhadap sistem. Sistem yang telah dibuat fokus pada keseluruhan fitur dan fungsional sistem. Sehingga sistem sesuai tujuannya dan dapat digunakan.

2.3 Pengertian Informasi

Informasi adalah sebuah data yang disajikan dalam bentuk yang berguna untuk membuat suatu keputusan (P. Miguel et al., 2014).

Informasi didefinisikan ialah sebagai hasil dari suatu pengolahan data dalam bentuk yang lebih berguna serta lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan sebuah keputusan (Meningkatkan & Perusahaan, n.d.).

Informasi adalah pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat ditafsirkan dari pesan atau kumpulan pesan (Solahudin, 2021).

Berdasarkan para ahli diatas dapat penulis mendapatkan kesimpulan bahwa informasi merupakan kumpulan data yang telah dikelola, diproses, dan siap digunakan untuk mendukung suatu keputusan.

2.4 Pengertian Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul ada terjadi (Yani et al., 2019).

Sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam sub sistem yang lebih kecil yang mendukung sistem yang lebih besar (Rivanthio et al., 2017).

Sistem adalah suatu kesatuan yang terdiri atas komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk menggambarkan suatu set entitas yang berinteraksi (Ningsih & sabandi, 2019).

2.5 Pengertian Sistem Informasi Akademik

Berdasarkan kutipan diatas penulis mendapatkan kesimpulan bahwa sistem merupakan kumpulan beberapa objek yang memiliki keterkaitan satu sama lain untuk mencapai tujuan dari sistem tersebut.

Sistem Informasi Akademik adalah aplikasi yang dirancang untuk kebutuhan pengolahan data administratif sekolah dengan tujuan supaya data akademik lebih terkelola dengan baik. Biasanya berhubungan dengan fitur seperti:

- a. Penyimpanan data siswa/i.
- b. Pembagian kelas.
- c. Jadwal pelajaran.
- d. Penilaian.
- e. Rekapitulasi hasil belajar.
- f. Perkembangan kemajuan belajar.
- g. Pelaporan hasil belajar siswa/i.
- h. Media pemberian informasi dari guru kepada siswa/i.

Sehingga dengan adanya pemanfaatan aplikasi seperti ini diharapkan setiap kegiatan akademik dapat terkelola dengan baik untuk memaksimalkan pelayanan akademik di lingkungan sekolah (Solahudin, 2021).

Sistem informasi akademik salah satu media yang membantu sekolah untuk mengelola data akademik. Sistem informasi akademik merupakan salah satu media utama dalam mengatur hal-hal yang menyangkut penyelenggaraan kegiatan, didalam sistem terdapat komponen-komponen yang saling berinteraksi. Suatu sistem informasi akademik mampu melakukan hal-hal yang berkaitan dengan penyelenggaraan maupun hal-hal spesifikasi lainnya, komponen-komponen

dipermudah dengan adanya sistem ini, tidak perlu terjadi kesalahpahaman jika aturan-aturan sudah masuk ke dalam sistem.

2.6 Flowchart Diagram

Flowchart adalah suatu bagan yang terdapat simbol-simbol tertentu untuk menggambarkan urutan proses secara mendetail dan menjelaskan secara singkat hubungan antara suatu proses (instruksi) dengan proses lainnya dalam program. Perancangan *flowchart* sebenarnya tidak butuh rumus atau patokan yang bersifat mutlak (pasti). Hal ini didasari oleh *flowchart* (bagan alir) adalah sebuah gambaran yang terbentuk dari hasil pemikiran dalam hal menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan lainnya. Secara garis besar setiap melakukan perancangan *flowchart* terdapat tiga bagian yang wajib diperhatikan, yaitu *input*, proses dan *output*. Berikut ini merupakan simbol-simbol dalam penggambaran *flowchart*:

2.7 Unified Modelling Language (UML)

UML merupakan singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Chonoles (2003) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika ingin membuat model menggunakan konsep UML terdapat peraturan yang harus diperhatikan. UML dapat diaplikasikan untuk maksud tertentu antara lain:


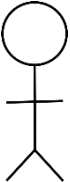
1. Merancang Perangkat Lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisis dan mencari hal-hal yang diperlukan sistem.
4. Mendokumentasi sistem yang terdapat suatu proses dan organisasinya.



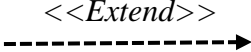
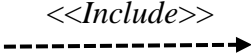
Pembangun utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis timing diagram) dan ada yang bersifat umum (misalnya diagram kelas). Pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang.

1.7.1 Use Case Diagram

Diagram *use case*, bersifat statis. Diagram memperlihatkan himpunan *use case* dan aktor (jenis khusus dari kelas). Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku sistem yang dibutuhkan serta diharapkan pengguna. Berikut ini merupakan simbol-simbol yang terdapat pada *use case* diagram:

Tabel 2.1 Simbol-simbol *Use Case* Diagram (Bakhri, 2019)

Simbol	Keterangan
	<p><i>Use Case</i> : Unit-unit atau <i>actor</i>, yang biasanya dinyatakan dengan kata kerja diawal frase <i>use case</i>.</p>
	<p>Aktor : Orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat selain dari sistem informasi yang dibuat itu sendiri, walaupun simbol dari aktor ini adalah berbentuk gambar orang, tapi aktor belum tentu orang, biasanya</p>



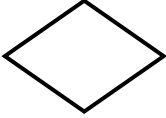

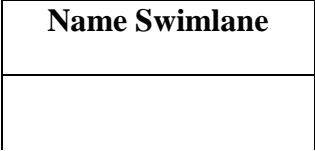

	dinyatakan menggunakan kata benda di awal frase aktor.
	Asosiasi : Sebuah komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	Generalisasi : Hubungan umum-khusus antara dua <i>use case</i> yang salah satu fungsinya adalah fungsi yang lebih umum.
	<i>Extend</i> : Simbol ini menunjukkan bahwa <i>use case</i> tambahan ke <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan tersebut.
	<i>Include</i> : Simbol ini menunjukkan bahwa <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.

1.7.2 Activity Diagram

Diagram aktivitas, bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran suatu aktivitas ke aktivitas lainnya dalam sistem. Diagram ini penting dalam suatu sistem untuk pemodelan fungsi suatu

sistem dan memberi tekanan aliran kendali antar objek. Berikut ini merupakan simbol-simbol yang terdapat pada *activity* diagram:




Tabel 2.2 Simbol-simbol *Activity* Diagram (Bakhri, 2019)




Simbol	Keterangan
	Simbol status awal : Status awal dari diagram aktivitas.
	Simbol Aktivitas : Aktivitas yang dilakukan oleh sistem yang diawali dengan kata kerja.
	Simbol <i>decision</i> : Asosiasi percabangan, dimana jika terdapat pilihan aktivitas yang lebih dari satu.
	Simbol <i>Join</i> : Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Simbol <i>swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
	Simbol status akhir : Status akhir dari diagram aktivitas.

1.7.3 Class Diagram

Diagram kelas, bersifat statis. Diagram yang memperlihatkan himpunan kelas kelas, antarmuka, kolaborasi-kolaborasi, serta relasi. Diagram ini sering ditemukan pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas aktif.. Berikut ini merupakan simbol-simbol yang terdapat pada *class* diagram:

Tabel 2.3 Simbol-simbol *Class* Diagram (Bakhri, 2019)

Simbol	Keterangan			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 2px;">Nama_Kelas</td> </tr> <tr> <td style="padding: 2px;">+ atribut</td> </tr> <tr> <td style="padding: 2px;">- operasi</td> </tr> </table>	Nama_Kelas	+ atribut	- operasi	Merupakan kelas yang terdapat pada struktur.
Nama_Kelas				
+ atribut				
- operasi				
	Asosiasi berarah (<i>Directed Association</i>) : Merupakan relasi antar kelas dengan maksud kelas yang satu digunakan oleh kelas yang lain, asosiasi juga disertai dengan <i>multiplicity</i> .			
	Generalisasi (<i>Generalisation</i>) : Merupakan relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).			
	Ketergantungan (<i>Dependency</i>) : Merupakan relasi antarkelas dengan makna ketergantungan antar kelas.			

	Agregasi (<i>Agregation</i>) : Merupakan relasi antarkelas dengan makna semua bagian (<i>whole-part</i>).
	Antar muka (<i>Interface</i>) : Sama seperti konsep <i>interface</i> yang terdapat pada pemrograman berorientasi objek
	Asosiasi (<i>Assosiation</i>): Merupakan relasi antarkelas dengan makna umum, asosiasi juga disertai dengan <i>multiplicity</i> .

2.8 PHP

PHP merupakan singkatan dari *PHP: Hypertext Processor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *Web* yang disisipkan pada dokumen *HTML*. Penggunaan *PHP* memungkinkan *Web* dapat dibuat dinamis sehingga maintenance situs *Web* tersebut menjadi lebih mudah dan efisien. *PHP* merupakan *software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di *download* secara bebas dari situs resminya <http://www.php.net> (Suhartanto, 2016).

PHP: Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML* (Raharjo, Heryanto dkk,2010). *PHP* adalah bahasa pemrograman *server side* yang sudah banyak digunakan pada saat ini, terutama untuk pembuatan *website* dinamis. Untuk hal-hal tertentu dalam pembuatan *web*,

bahasa pemrograman *PHP* memang diperlukan, misalnya saja untuk memproses data yang dikirimkan oleh pengunjung *web* (maharani, 2018).

2.9 MySQL

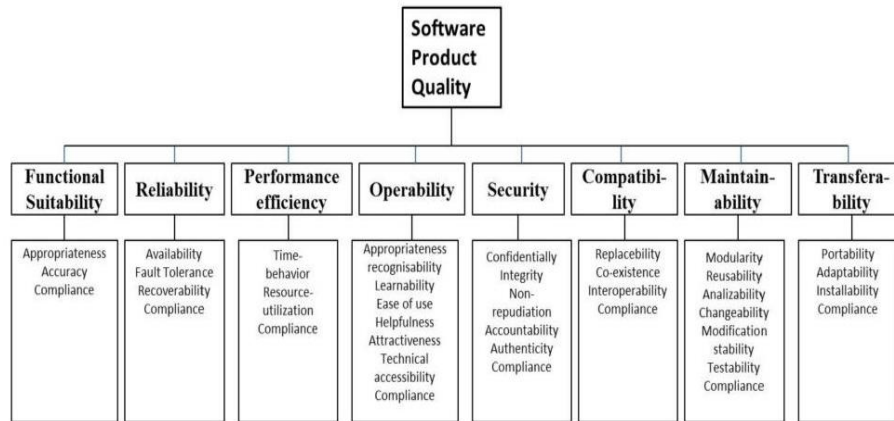
Menurut Rochman,dkk (2018), *MySQL* didefinisikan sebagai nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah pengorganisasian data. Menggunakan *MySQL*, bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah.

Menurut (Suhartanto, 2018), *MySQL* adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis *website* yang dinamis. *MySQL* termasuk jenis RDBMS (*Relational Database Management System*). *MySQL* ini mendukung bahasa pemrograman *PHP*. *MySQL* memiliki *Query* atau bahasa *SQL* (*Structured Query Language*) yang *simple* dan menggunakan *escape character* yang sama dengan *PHP*.

Berdasarkan pengertian diatas dapat disimpulkan bahwa *MySQL* merupakan salah satu jenis RDBMS (*Relational Database Management System*) yang salah satu aplikasi sistem berfungsi untuk menjalankan pengolahan data dan membuat *website* yang bersifat dinamis.

2.10 Pengujian ISO 25010

Menurut (Borman et al., 2020), ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer, yang dapat terdapat pada gambar dibawah ini:



Gambar 2.2 Model ISO 25010 (Borman et al., 2020)

Berdasarkan gambar diatas, dapat dijelaskan mengenai delapan karakteristik tersebut, sebagai berikut:

1. *Functional Suitability*, merupakan sistem atau produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.

2. *Reliability*, merupakan tingkat dimana suatu sistem atau produk dapat mempertahankan kinerjanya pada tingkat tertentu ketika digunakan pada keadaan tertentu.

3. *Performance Efficiency*, merupakan tingkat dimana sistem atau produk menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.

4. *Usability*, merupakan tingkat dimana pada suatu sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.

5. *Security*, merupakan tingkat dimana pada suatu sistem atau produk menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, ataupun pengungkapan yang berbahaya.

6. *Compatibility*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.

7. *Maintainability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.

8. *Portability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dipindahkan dari satu ruang ke ruang lainnya.

2.11 PIECES

Menurut (Susanto & Ayu, 2017), metode *PIECES* merupakan sebuah metode analisis yang digunakan sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Analisis ini disebut dengan Analisis *PIECES* (*Performance, Information, Economy, Control, Efficiency, and Service*).

2.12 Drawio

Drawio adalah editor grafis vektor dan alat *prototyping* dengan berbasis *web* serta fitur *offline* tambahan yang diaktifkan oleh aplikasi *desktop* untuk *Mac OS* dan *Windows*. Aplikasi pendamping *drawio* untuk *Android* dan *iOS* memungkinkan untuk melihat *prototype* *drawio* pada perangkat seluler. Rangkaian fitur *drawio* berfokus pada penggunaan dalam antarmuka pengguna dan desain pengalaman pengguna dengan penekanan pada kolaborasi waktu nyata (*real-time*).

2.13 PHP Native

Menurut (Fadhla Binti Junus, 2020) *PHP Native* sendiri merupakan pembuatan website dinamis dengan php yang prosedural dimana pengerjaannya dimulai dari 0 tanpa menggunakan *framework* maupun *plugin* yang tersedia. *PHP* pertama kali dibuat dan diperkenalkan oleh Rasmus Lerdorf pada tahun 1995 menggunakan nama *PHP/FI*, generasi awal *PHP/FI* dibuat dari *Perl* yang waktu itu digunakan untuk kebutuhan pribadi saja.

