# BAB II TINJAUAN PUSTAKA

* 1. **Landasan Teori**
     1. **Sistem Informasi**

Data bisa didefinisikan seperti benda, fakta kejadian, dan transaksi yang terjadi tetapi tidak memiliki pengaruh langsung kepada pihak pemakai atau transaksi yang tidak memiliki makna. Data dapat berupa teks, audio, dan video (Liana, Sutardi, & Muchlis, 2018).

Informasi adalah kumpulan-kumpulan data yang diolah sehingga menjadi sebuah informasi yang memiliki makna bagi si penerima informasi tersebut (Lestariningsih, dkk, 2016).

Menurut Laudon, Sistem Informasi adalah secara teknis sebagai suatu rangkaian yang komponen-komponennya saling berkaitan dalam mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan mengendalikan perusahaan (Ardana, Lukman, 2016).

Berdasarkan definisi diatas penulis dapat menyimpulkan bahwa sistem informasi merupakan komponen yang saling berkaitan dalam mengumpulkan, memproses, menyimpan serta melaporkan informasi untuk menunjang pengambilan keputusan demi mencapai tujuan perusahaan.

* + 1. **Program Magang Mahasiswa Bersertifikat**

Program Magang Mahasiswa Bersertifikat (PMMB) 2020 adalah hasil kerja sama antara Kementerian Pendidikan dan Kebudayaan dan Kementerian Badan Usaha Milik Negara melalui. Peluncuran PMMB tahun 2020 ini dibuka oleh Menteri Badan Usaha Milik Negara (BUMN), Erick Thohir dan dihadiri oleh Menteri Pendidikan dan Kebudayaan, serta turut dihadiri direksi dari 143 Badan Usaha Milik Negara (BUMN) dan mahasiswa serta rektor dari 300 Perguruan Tinggi Negeri/Swasta dari seluruh Indonesia (Kemendikbud, 2020).

Maksud dari Program Magang Mahasiswa Bersertifikat (PMMB) adalah Untuk menjawab tantangan Sumber Daya Manusia Badan Usaha Milik Negara (BUMN) dimasa yang akan datang, menciptakan Sumber Daya Manusia unggul dengan kompetensi yang mumpuni melalui Pemagangan di Badan Usaha Milik Negara (BUMN) dan Mencetak Sumber Daya Manusia yang berdaya saing global (FHCI, 2021).

Dan Tujuan dari Program Magang Mahasiswa Bersertifikat (PMMB) adalah Menghadirkan Sinergi melalui Badan Usaha Milik Negara (BUMN) Hadir Untuk Negeri, Kandidat rekrutment untuk Badan Usaha Milik Negara (BUMN) terkait, dan Menciptakan Sumber Daya Manusia yang berjiwa Entreprenuer sesuai dengan kebutuhan Badan Usaha Milik Negara (BUMN) (FHCI, 2021).

Adapun Jenis Program Magang Mahasiswa Bersertifikat (PMMB) yang disebutkan oleh Forum Human Capital Indonesia (FHCI) ada 2 yaitu:

1. Magang Bersertifikat Industri

Mahasiswa melaksanakan pemagangan sesuai dengan Project yang diberikan oleh pihak Industri selama 6 (enam) bulan, Peserta magang dinyatakan selesai mengikuti pemagangan setelah menyelesaikan pekerjaan dengan waktu yang telah di tentukan, yang kemudian di akui oleh pihak Industri dengan mengeluarkan Sertifikat Industri sesuai dengan project.

1. Magang Bersertifikat Kompetensi

Mahasiswa melaksanakan pemagangan sesuai dengan kompetensi Bidang (Posisi) yang menjadi persyaratan untuk menduduki posisi tsb selama 6 (enam) bulan, Peserta Magang akan melaksanakan Uji Kompetensi sesuai dengan Kompetensi Bidang pada akhir masa pemagangan yang dilakukan oleh Lembaga Sertifikasi Profesi Nasional

* + 1. ***Website***

Website terhubung dengan suatu jaringan internet yang akan membawa pengguna kesuatu tujuan yang diinginkan oleh pengguna dengan cara mengklik link yang berupa teks, gambar (Endra & Aprilita, 2018).

Menurut Ardhana dalam zamaludin, website adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan surfer (sebutan pemakai komputer yang melakukan browsing atau penelusuran informasi melalui internet) (Zamaludin, 2016).

Dari pengertian diatas penulis menyimpulkan website adalah sebuah kumpulan layanan informasi yang berisikan sebuah domain yang mengandung informasi, aplikasi yang menggunakan konsep hyperlink untuk memudahkan para pengguna.

* + 1. **PT Timah Tbk**

**PT TIMAH Tbk sebagai Perusahaan Terbatas TIMAH didirikan pada 2 Agustus 1976. Ini adalah Badan Usaha Milik Negara yang bergerak di bidang penambangan timah dan telah terdaftar di Bursa Efek Indonesia sejak 1995.**

PT TIMAH Tbk merupakan produsen dan eksportir logam timah, dan memiliki segmen usaha penambangan timah terintegrasi mulai dari kegiatan eksplorasi, penambangan, pengolahan hingga pemasaran. Ruang lingkup kegiatan Perusahaan meliputi juga bidang pertambangan, perindustrian, perdagangan, pengangkutan dan jasa. Kegiatan utama perusahaan adalah sebagai perusahaan induk yang melakukan kegiatan operasi penambangan timah dan melakukan jasa pemasaran kepada kelompok usaha mereka. Perusahaan memiliki beberapa anak perusahaan yang bergerak dibidang perbengkelan dan galangan kapal, jasa rekayasa teknik, penambangan timah, jasa konsultasi dan penelitian pertambangan serta penambangan non timah (Timah, 2021).

Perusahaan berdomisili di Pangkalpinang, Provinsi Bangka Belitung dan memiliki wilayah operasi di Provinsi Kepulauan Bangka Belitung, Provinsi Riau, Kalimantan Selatan, Sulawesi Tenggara serta Cilegon, Banten.



**Gambar 2.1.** Logo PT Timah Tbk (Timah, 2024)

* + 1. ***Agile System Development Life Cycle***

Menurut akbar dkk, *Agile development* adalah sebuah konsep pengembangan yang memiliki arti cepat dilakukan, cepat merespon perubahan yang diminta oleh klien, dan melibatkan secara secara aktif dalam proses sehingga produk yang dihasilkan atau perangkat lunak yang dikembangkan merupakan hasil dari setiap pihak yang terlibat (Akbar, 2017).

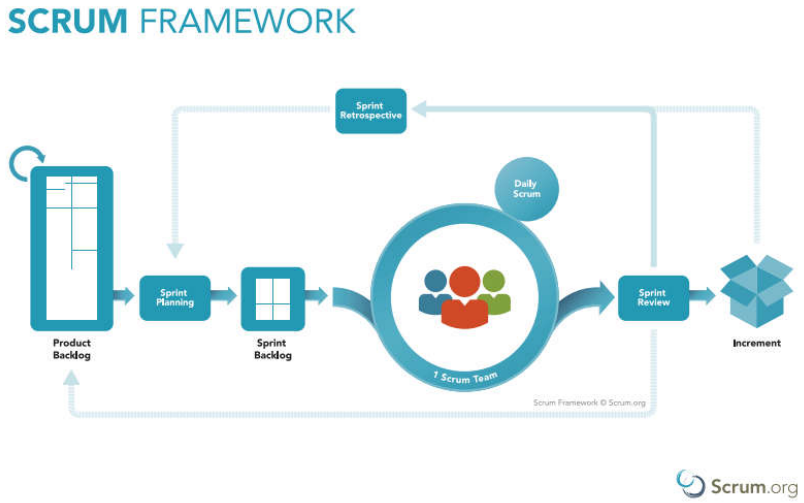
Menurut Suryantara, metodologi *agile* merupakan metodologi yang cepat. Beberapa metodologi *agile* yang berkembang saat ini antara lain (Suryantara, 2017):

1. *Extreme Programming* (XP)
2. *Scrum Methodology*
3. *Crystal Family*
4. *Dynamic Systems Development Method* (DSDM)
5. *Adaptive Software Development* (ASD)
6. *Feature Driven Development* (FDD)

Dari kedua pendapat tersebut bisa disimpulkan bahwa metodologi agile merupakan metodologi pengembangan perangkat lunak didasarkan pada prinsip-prinsip yang sama. Metodologi *agile* ini bisa melaksanakan sesuatu pengembangan sistem secara jangka pendek dengan membutuhkan menyesuaikan diri yang kilat dari pengembang terhadap pergantian dalam wujud apapun. Sehingga metodologi *agile* ini telah berkembang menjadi beberapa metodologi seperti *Extreme Programming* (XP), *Scrum Methodology*, *Crystal Family*, *Dynamic Systems Development Method* (DSDM), *Adaptive Software* *Development* (ASD), serta *Feature Driven Development* ( FDD).

* + 1. ***Scrum Framework***

Menurut schwaber dan Sutherland dalam ginanjar, *Scrum* adalah sebuah framework di mana orang-orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, dimana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif (Ginanjar, 2017).



**Gambar 2.2.** Scrum Framework (Scrum.org, 2017)

* + - 1. **Tim *Scrum***

Tim dalam *Scrum* terdiri dari 3 bagian, yaitu *Product Owner*, tim pengembang, dan *Scrum* master (Ginanjar, 2017). Di dalam *Scrum*, setiap tim harus bisa mengatur dirinya sendiri sehingga tim bisa menyelesaikan kerjanya sesuai dengan kebutuhan masing-masing.

1. *Product Owner*

Menurut Rubin dalam Ginanjar, *Product Owner* adalah pemimpin utama sebuah produk. Dia adalah Otoritas tunggal yang bertanggung jawab untuk menentukan fitur dan fungsionalitas mana yang akan dibangun dan perintah untuk membangunnya (Ginanjar, 2017). *Product Owner* juga memiliki tanggung jawab untuk mengelola *Product Backlog* dan memastikan *Product Backlog* dikerjakan sesuai dengan perencanaan.

1. Tim Pengembang

Tim Pengembang terdiri dari para profesional yang bekerja untuk menghasilkan produk (Ginanjar, 2017).

Dalam Panduan *Scrum* yang ditulis oleh Schwaber dan Sutherland, Tim Pengembang memiliki karakteristik sebagai berikut :

1. Mengatur dirinya sendiri untuk menyelesaikan potongan *Product Backlog*.
2. Tim Pengembang memiliki keahlian yang dibutuhkan selama pengembangan produk.
3. Tidak ada jabatan tertentu dalam tim pengembang.
4. Tim pengembang tidak mengenal adanya sub-tim yang dikhususkan untuk bidang tertentu seperti pengajuan atau analisis bisnis.
5. Anggota tim boleh memiliki keahlian tertentu dan fokus dalam satu area tertentu, namun akuntabilitas dari hasil pekerjaan secara keseluruhan adalah milik tim pengembang.
6. *Scrum Master*

Menurut Scwaber dan Sutherland dalam Ginanjar, *Scrum Master* adalah seseorang yang bertanggung jawab untuk memastikan *Scrum* telah dipahami dan dilaksanakan. *Scrum* *Master* adalah seorang pemimpin yang melayani Tim *Scrum*

* + - 1. **Kegiatan *Scrum***

Kegiatan dalam *Scrum* yang biasanya disebut *Sprint*. *Sprint* adalah sebuah batasan waktu untuk menyelesaikan sebuah potongan produk. Tahapan selama satu *sprint* terdiri dari *Sprint Planning, Daily Scrum, Development, Sprint Review, Sprint Retrospective* (Ginanjar, 2017).

1. *Sprint Planning*

Sprint planning adalah kegiatan dimana tim *Scrum* merencakan apa yang akan dikerjakan dalam satu sprint kedepan. Pada tahapan Sprint Planning setiap *developer* harus paham dengan pencapaian apa yang harus selesai selama *Sprint* berlangsung. Hasil dari sprint planning disajikan dengan table sprint produk *backlog* dan tabel *setup*. Tabel sprint backlog adalah daftar terurut Product *Backlog item* yang akan dikerjakan selama pelaksanaan sprint. Tabel *sprint backlog* akan banyak dibahas pada bagian artefak *scrum*. Tabel *setup* digunakan untuk mencatat nilai estimasi awal untuk acuan dalam pelaksanaan *sprint*. Tabel 2.1. merupakan contoh dari table *setup*.

**Tabel 2.1.** Contoh Tabel Setup Sprint

|  |  |
| --- | --- |
| **Sprint 1** | **Nilai** |
| Tanggal Mulai | 11 Oktober 2016 |
| Tanggal Berakhir | 27 Oktober 2016 |
| Jumlah *Developer* | 1 |
| *Focus Factor* | 0.7 |
| *Work Days Total* | 14 |
| *Man Days Total* | 14 |
| Jumlah *Story Point* | 10 |
| *i-deal task days left* | -0.714 |
| *i-man days used* | 1 |

Keterangan untuk Tabel 2.1 adalah sebagai berikut :

1. Tanggal mulai adalah tanggal dimulainya *sprint*.
2. Tanggal akhir adalah tanggal berakhirnya *sprint*.
3. Jumlah *developer* adalah jumlah orang dalam tim yang akan mengerjakan *sprint*.
4. *Focus Factor* adalah nilai *focus factor* yang digunakan pada *sprint* yang biasanya didapat dari nilai *focus factor* sebelumnya atau menggunakan nilai *default* 0,7 untuk tim yang baru melaksanakan *sprint* .
5. *Work days* total adalah total jumlah hari kerja pada satu *sprint*.
6. *Man days* total adalah nilai dari *work days* total dikali dengan jumlah *developer*.
7. *i - Ideal task days left* adalah nilai *ideal story point* yang harus berkurang untuk setiap harinya. Nilai *ideal task days left* didapat dari jumlah *story point* dibagi dengan *work days total*. Nilai *ideal task day left* akan digunakan untuk membuat garis lurus *ideal* *task days left* pada grafik *burndown chart*.
8. *i-Man days used* adalah nilai dari jumlah hari kerja yang dipakai oleh tim untuk setiap harinya.
9. *Daily Scrum*

*Daily Scrum* adalah kegiatan dengan batasan waktu 10 menit setiap harinya agar tim bisa menyingkronkan pekerjaan dan merencanakan apa yang akan dikerjakan di hari selanjutnya.

1. *Development*

Pada tahapan ini Tim mulai melakukan pekerjaan untuk menghasilkan sebuah produk sesuai dengan hasil *Sprint Planning*.

1. *Sprint Review*

*Sprint Review* adalah kegiatan yang dilakukan disetiap akhir *sprint* untuk meninjau potongan produk yang sudah dikerjakan dan belum selesai dikerjakan.

1. *Sprint Retrospective*

*Sprint Retrospective* adalah sebuah kesempatan bagi tim *Scrum* untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

* + 1. **WWW (*World Wide Web*)**

*World Wide Web* adalah suatu program yang ditemukan oleh Tim Barners-Lee pada tahun 1991 untuk menyusun arsip-arsip risetnya, lalu beliau mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah *program* piranti lunak yang diberi nama *Enquire*. Dengan *program* itu Barners-Lee berhasil menciptakan jaringan yang menautkan berbagai arsip sehingga memudahkan pencarian informasi yang dibutuhkan. Inilah yang kelak menjadi dasar dari sebuah perkembangan pesat yang dikenal dengan WWW (Hidayatullah dan Kawistara, 2017).

WWW bekerja berdasarkan pada tiga mekanisme berikut (Hidayatullah dan Kawistara, 2017):

1. Informasi disimpan didalam dokumen yang sering kita sebut halaman *web*.
2. Halaman *web* adalah *file-file* yang disimpan dalam komputer. Komputer tersebut dikenal dengan istilah *web server*.
3. Komputer yang mengakses isi dari halaman *web* disebut dengan *web client*.
4. *Web client* menampilkan halaman *web* dengan *program* yang dikenal dengan nama *web browser* seperti *Chrome, Firefox, dan Internet Explorer*.
   * 1. ***Web Browser***

Menurut Kasiman dalam susanti, *web browser* adalah *program* untuk menampilkan halaman yang berbentuk kode *HTML*. Semua halaman *web* ditulis dengan Bahasa *HTML* (*Hypertext Mark Up Language*). Walaupun berbeda *file* mempunyai ekstensi yang berbeda adalah *medium* yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman statis, sebuah *script* (seperti *PHP*), ataupun yang dibuat oleh program CGI (*Common Gateway Interface*) (Susanti, 2016).

* + 1. ***Web Server***

*Web server* merupakan perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan kembali ke browser. Contoh *Web server*, adalah IIS (*Internet Information Service*) Produk *Microsoft Corf* (Yunita, 2017).

*Web Server* adalah tempat dimana anda penyimpan aplikasi *web* anda kemudian mengaksesnya melalui Internet. Setiap perubahan kecil maupun besar anda upload ke *web server* setelah itu anda periksa apakah perubahan itu susuai yang anda inginkan atau belum (Hidayatullah dan Kawistara, 2017).

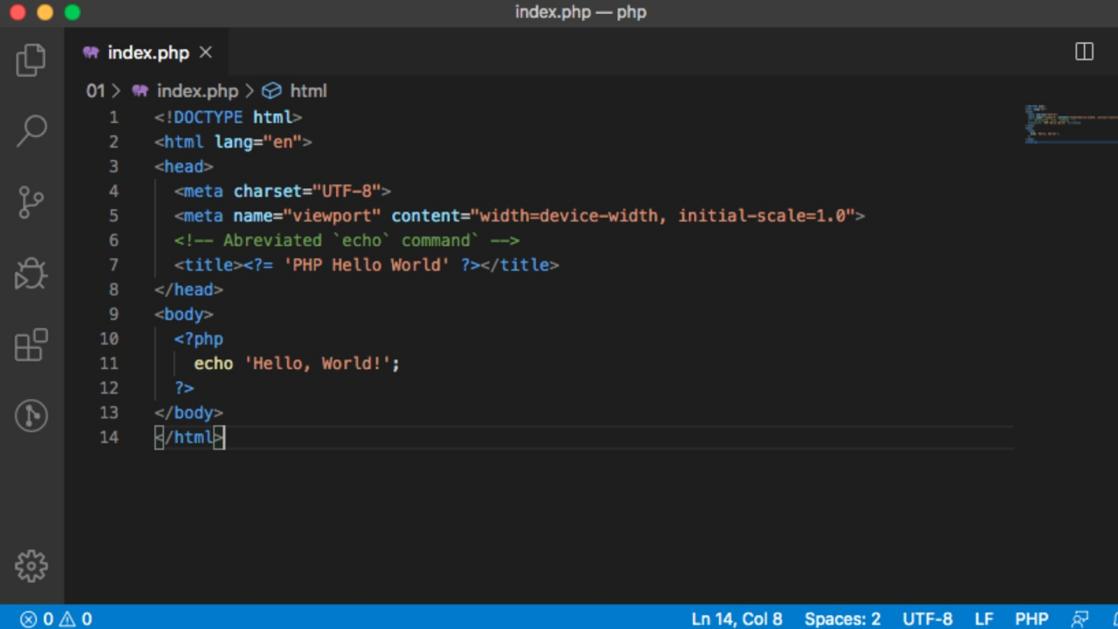
Berdasarkan teori diatas, *web server* sebuah komputer yang terdiri dari perangkat lunak dan perangkat keras yang bisa mendukung berbagai protokol *web*, seperti *HTTP, HTTPS*, dan lain-lain untuk memproses permintaan *client*.

* + 1. **PHP (*Hypertext Preprocessor*)**

PHP (*Hypertext Preprocessor*) merupakan Bahasa scripting yang sangat diketahui sering dipakai dalam pengembangan *web* walaupun sebenarnya tidak hanya itu kegunaannya (Das & Saikia, 2016).

PHP adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam HTML (Supono & Putratama, 2016).

Sedangkan menurut Solichin, PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web* (Solichin, 2016).



**Gambar 2.3.** Contoh Source Code PHP

Dari pendapat diatas penulis menyimpulkan bahwa *hypertext preprocessor* (PHP) merupakan bahasa pemrograman untuk membuat/mengembangkan aplikasi berbasis web dan bersifat *open soure* dan ditanamkan ke dalam *script* HTML.

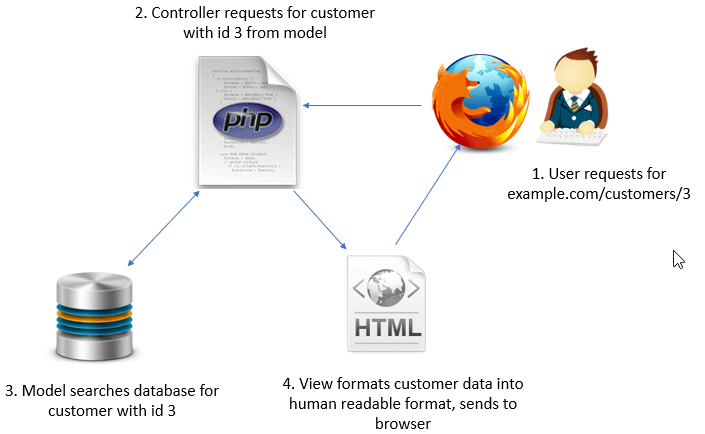
* + 1. ***Framework Codeigniter***

Menurut Betha Sidik dalam Destiningrum dan Adrian, *Framework* adalah kumpulan instruksi-instruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpat harus menuliskan *syntax* *program* yang sama berulang-ulang serta dapat menghemat waktu (Destiningrum & Adrian, 2017).

Codeigniter adalah sebuah *framework* PHP yang bersifat *open source* dan menggunakan metode MVC (*Model, View Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuat dari awal (Destiningrum & Adrian 2017).

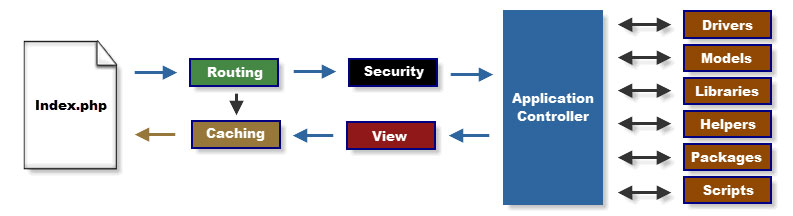
Codeigniter merupakan konsep MVC (*Model View Controller*) yang memungkinkan pemisahan antara *layer* *application logic* dan *presentation*. Dengan konsep ini kode PHP, Query, MySQL, Javacript, Dan CSS dapat saling dipisah-pisahkan sehingga ukuran *file* menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

1. *Model* menerapkan kode *program* (berupa OOP *class*) yang digunakan untuk berhubungan dengan *database* MySQL sekaligus untuk memanipulasinya (input-*edit*-*delete*).
2. *View* merupakan kode program berupa *template* atau PHP untuk menampilkan data pada *browser*.
3. *Controller* merupakan kode *program* (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol *model* dan *view*.



**Gambar 2.4.** Konsep Model View Controller (Techfor.id, 2020)

Adapun alur dari aplikasi yang ditulis menggunakan CodeIgniter seperti pada Gambar 2.5.



**Gambar 2.5.** Alur Codeigniter (Codeigniter.com, 2021)

1. *File* index.php, berfungsi sebagai *front controller*, menginisialisasi *resource* utama yang dibutuhkan untuk menjalankan CodeIgniter.
2. *Router*, memeriksa HTTP request untuk menentukan apa yang harus dilakukan.
3. Jika *file cache* ada, dikirim langsung ke *browser*, melewati eksekusi sistem normal.
4. Keamanan, sebelum *controller* aplikasi dimuat, HTTP *request* dan setiap data pengguna yang di*submit* disaring terlebih dahulu untuk keamanan.
5. *Controller*, memuat *model*, *library* utama, *helper*, dan setiap *resource* lainnya yang diperlukan untuk memproses permintaan khusus.
6. *View*, proses *render* kemudian dikirim ke *web browser* agar dapat dilihat. Jika *caching* diaktifkan, view di*cache* terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.
   * 1. **Basis Data**

Menurut Fathansyah dalam Winda Widiati menjelaskan bahwa Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya (Widiati, 2017).

*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur *file* yang menyusun sebuah *database* adalah Data *Record* dan *Field* (Yulia, 2017).

Dari pendapat diatas penulis menyimpulkan bahwa basis data adalah media untuk menyimpan data yang mana merupakan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* dan kolom.

* + 1. **MySQL**

Menurut Kurniawan, MySQL adalah salah satu jenis *database server* yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Management System*). MySQL ini mendukung bahasa pemrograman PHP, MySQL dan mempunyai bahasa SQL (*Structured Query Language*) yang simpel dan menggunakan *escape character* yang sama dengan MySQL, mempunyai tampilan *Client* yang mempermudah dalam mengakses *database* dengan kata sandi untuk mengijinkan proses yang boleh di lakukan. Untuk masuk ke dalam database disediakan *userdefault* yaitu *root* dengan *password* yang telah dibuat pada saat proses instalasi, yaitu *maxikom* (Dewi, Kurniati, & Irmayani, 2017).

MySQL merupakan *software* *database* yang paling populer dikarenakan performa *query* dari *database* yang bisa dikatakan paling cepat, dan bisa dibilang jarang bermasalah. Mulai dari versi 3.23 MySQL menjadi software *open source* yang berarti gratis, dapat digunakan untuk kepentingan komersial atau personal. MySQL kini dapat digunakan di Windows, yang pada awalnya digunakan di Linux. MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management System* atau RDBMS), seperti halnya ORACLE, *postgresql*, MS SQL dan sebagainya (Wisky, 2017).

Dari pendapat diatas penulis menyimpulkan bahwa MySQL adalah MySQL merupakan software *database* yang paling populer dikarenakan performa *query* dari *database* yang bisa dikatakan paling cepat, dan bisa dibilang jarang bermasalah.

* + 1. **UML (*Unified Manual Langauage*)**

Menurut Fowler, M. dalam B. O. Lubis, UML (*Unified Modeling Language*) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientansi objek (OO). Definisi ini merupakan definisi yang sederhana (B. O. Lubis, 2016).

Menurut Nugroho dalam Carolina dan Supriyatna, UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Carolina & Supriyatna, 2019)

Jadi dari pendapat diatas, UML (*Unified Modeling Language*) dapat diartikan sebagai bahasa *visual* untuk menggambarkan definisi-definisi tentang *requirement*, membuat analisis dan desain serta menggambar arsitektur dalam pemrograman berorientasikan objek dengan menggunakan teks-teks pendukung.

UML terdiri dari 13 macam diagram (Rosa & Salahuddin, 2016), namun penulis hanya menggunakan beberapa diagram, diantaranya:

1. *Use Case Diagram*

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut Aktordan *use case*.

1. Aktormerupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2.2.** Simbol Use Case Diagram

| **Simbol** | **Deskripsi** |
| --- | --- |
| *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau Aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal *frase* nama *use case* |
| *Actor*  Hasil gambar untuk Simbol Actor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari Aktor adalah gambar orang, tapi belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal *frase* nama *actor* |
| Asosiasi/*association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor* |
| Extensi/*extend*  *<<extend>>* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use* *case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misalnya:    Arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya |
| Generalisasi/ *generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :    Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum) |
| Menggunakan/*include/uses*  *<<include>>*  *<<uses>>* | Relasi *use case* tambahan ke sebuah *use c*ase dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai include di *use case* :   * Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, Misal pada kasus berikut:      * *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use* *case* tambahan dijalankan, misal pada kasus:     Kedua interprestasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interprestasi yang dibutuhkan. |

1. *Activity Diagram*

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Diagram aktivitas juga banyak digunakan untuk mendefinisi hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kamus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada *activity diagram*:

**Tabel 2.3.** Simbol Activity Diagram

| **Simbol** | **Deskripsi** |
| --- | --- |
| Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan / *decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / *join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| *Swimline*   |  | | --- | | **Nama Swimline** | |  |   Atau   |  |  | | --- | --- | | **Nama Swimline** |  | | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

1. *Class Diagram*

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 pola dan metode atau operasi:

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

Berikut adalah simbol-simbol yang ada pada *class diagram*:

**Tabel 2.4.** Simbol Class Diagram

| **Simbol** | **Deskripsi** |
| --- | --- |
| Kelas/*class*   |  | | --- | | Nama\_kelas | | + atribut | | - operasi | | Kelas pada struktur system |
| Antarmuka/*interface*  Nama\_interface | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| Asosiasi/*association* | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| Asosiasi berarah/*directed association* | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| Generalisasi/ *generalization* | Relasi antarkelas dengan makna generalisasi-speasialisasi (umum – khusus) |
| Kebergantungan / *dependency* | Kebergantungan antar kelas |
| Agregasi / *aggregation*  Hasil gambar untuk simbol Agregasi / aggregation Class Diagram | Relasi antarkelas dengan makna semua-bagian (*whole – part*) |

1. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Berikut adalah simbol-simbol yang ada pada *sequence diagram*:

**Tabel 2.5.** Simbol Sequence Diagram

| **Simbol** | **Deskripsi** |
| --- | --- |
| Aktor/*actor*  Hasil gambar untuk Simbol Actor  Nama Aktor  Atau  Nama Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari Aktor adalah gambar orang, tapi belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal *frase* nama Aktor |
| Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| Objek  Nama objek : nama kelas | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya    Maka cekStatusLogin() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif |
| *Create*  *<<create>>* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| Pesan tipe *call*  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 |
| Pesan tipe *send*  1 : masukan | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek yang lainnya. Arah panah mengarah pada objek yang dikirimi |
| Pesan tipe *return*  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 |
| Pesan tipe *destroy* | Menyatakan suatu objek mengahiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* |

* + 1. **ERD (*Entity Relationship Diagram*)**

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya.(Fridayanthie & Mahdiati, 2016).

ERD (*Entity Relationship Diagram*) adalah suatu *model* untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional (Rosa & Salahuddin, 2016).

**Tabel 2.6.** Simbol Entity Relationship Diagram

| **Simbol** | **Keterangan** |
| --- | --- |
| Entitas/*Entity*  Nama\_entitas | Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi *computer*; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| Atribut | *Field* atau kolom data yang butuh disimpan dalam suatu entitas. |
| Atribut kunci primer | *Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama). |
| Atribut multinilai/ *multivalue* | *Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. |
| Relasi  Nama\_relasi | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja. |
| Asosiasi / *association* | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B. |

* + 1. ***Penetration Testing***

Penetration Testing Merupakan sebuah kegiatan yang dilakukan untuk menguji sebuah kerentanan keamanan pada sebuah aplikasi dan mengetahui resiko yang akan terjadi kedepannya (Yunanri, 2018). Dan dibawah ini terdapat 3 model *pentest* sebagai berikut:

1. *Black Box*

*Black box* merupakan sebuah tipe pengujian yang lebih berfokus terhadap sudut pandang external, disini pengujian hanya mencari input dan *output* dari sebuah sistem (Robertus, 2017).

1. *White Box*

Menurut Robertus, *White box* merupakan sebuah tipe pengujian yang lebih berfokus terhadap sudut pandang internal, disini pengujian terhadap kesalahan terhadap fungsional yang spesifik (Robertus, 2017).

1. *Gray Box*

Menurut Robertus, *Gray box* merupakan sebuah tipe pengujian yang didapatkan dari kombinasi dari *black box* dan *white box*, disini pengujian tersebut menggunakan *framework* yang digunakan oleh aplikasi (Robertus, 2017).

* 2. **Penelitian Terdahulu**

**Tabel 2.7.** Penelitian Terdahulu

| **No** | **Judul** | **Peneliti, Media Publikasi, dan Tahun** | **Tujuan Penelitian** | **Kesimpulan** | **Saran atau Kelemahan** | **Perbandingan** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Sistem Monitoring Kegiatan Kerja Praktik Studi Kasus Program Studi Teknik Informatika S1 Universitas Muhammadiyah Magelang | Achmad Syaifur Rochman, Skripsi Teknik Informatika Universitas Muhammadiyah Magelang, 2020 | Merancang sistem informasi kegiatan kerja praktik mahasiswa untuk mempermudah mahasiswa menginputkan data kegiatan dan mempermudah dosen pembimbing untuk mengelola data kegiatan mahasiswa. | Sistem informasi KP memudahkan mahasiswa menginputkan data kegiatan KP dan foto KP ke sistem sebagai laporan selama selama kerja praktik dan memudahkan dosen pembimbing memantau kegiatan mahasiswa selama di tempat KP.  Sistem ini dapat menampilkan secara detail kegiatan mahasiswa di tempat KP dan juga dosen pembimbing dan pihak Instansi/perusahaan nantinya akan dengan mudah memverifikasi data kegiatan KP di sistem. | Meningkatkan keamanan data agar tidak mudah di ubah oleh pihak-pihak yang tidak bertanggung jawab. | Metode penelitian yang digunakan adalah model *waterfall.*  Tidak ada penilaian terhadap kinerja mahasiswa nya |
| 2 | Sistem Informasi Monitoring Peserta Praktik Kerja Lapangan Pada PDAM Surya Sembada Surabaya | Khairil, Feronika, Rahayu, Hanna, Tria, Jurnal Skripsi Ilmu Komputer Universitas Pembangunan Nasional, 2020 | Membantu tugas pegawai Tata Usaha untuk mempermudah melakukan pengelolaan data kegiatan PKL dengan mudah dan lebih terstruktur, membantu pembimbing lapangan dalam melihat *report* dengan jangka waktu tertentu. Pengembangan Sistem Informasi PKL ini diharapkan dapat membantu PDAM Surya Sembada Kota Surabaya dalam meningkatkan kinerjanya. | PDAM Surya Sembada kota Surabaya membutuhkan sistem informasi untuk menyimpan data peserta kerja praktek lapangan dan memberikan informasi secara mudah bagi pimpinan maupun calon peserta yang akan melakukan pengajuan permohonan kerja praktek di PDAM Surya Sembada kota Surabaya dengan dibuatnya sebuah sistem informasi berbasis website.  Pengelolaan data oleh admin menjadi lebih mudah dan efektif untuk menambahkan data dan juga mengganti data dari sistem informasi. | Meningkatkan keamanan untuk menjaga keamanan data.  pengembangan sistem dapat dilakukan agar sistem informasi dapat diakses melalui Smartphone/Android. | *Database Management System* (DBMS) yang digunakan sistem ini menggunakan *Postgresql*.  Metode penelitian yang digunakan menggunakan model *waterfall*  Tidak ada penilaian terhadap kinerja mahasiswa nya  Tidak adanya fitur pemberian tugas kepada mahasiswa yang akan magang |
| 3 | Rancang Bangun Aplikasi Penerimaan Mahasiswa Magang Pada Dinas Komunikasi Dan Informatika Provinsi Jawa Timur Berbasis Website | Dika Arissa Santoso Putri, Skripsi Teknologi Dan Informatika Universitas Dinamika, 2020 | Membuat sebuah aplikasi pengajuan magang pada Dinas Komunikasi dan Informatika Provinsi Jawa Timur Berbasis Website | Aplikasi yang dibangun dapat mempermudah mahasiswa dalam mengajukan permohonan magang di Dinas Komunikasi dan Informatika Provinsi Jawa Timur  Aplikasi yang dibangun mempermudah admin dalam melakukan pendataan mahasiswa yang mengajukan magang dan pernah magang di Dinas Komunikasi dan Informatika Provinsi Jawa Timur  Pembangunan aplikasi berbasis objek menggunakan framework memudahkan pengembang dari segi penkodean. Selain itu modifikasi terhadap aplikasi juga lebih mudah karena menyesuaikan dengan kerangka yang telah ada. | Pada aplikasi penerimaan mahasiswa pada Dinas Komunikasi dan Informatika Provinsi Jawa Timur diharapkan dapat berkembang dengan penambahan fitur-fitur seperti presensi  Aplikasi dapat diintegrasikan dengan aplikasi lainnya yang ada pada PT Jasaraharja Putera | Metode penelitian menggunakna model waterfall  Tidak ada penilaian terhadap kinerja mahasiswa nya  Tidak adanya fitur pemberian tugas kepada mahasiswa yang akan magang |
| 4 | Perancangan Sistem Informasi Program Pemagangan Dan Persebaran Perusahaan Tempat Peserta Magang Di Jepang Pada LPK Mulia Mandiri Indonesia | Titan Meiyansyah, Skripsi Sistem Informasi Universitas Widyatama, 2018 | Untuk merancang sistem informasi program pemagangan dan persebaran perusahaan tempat peserta magang di Jepang studi kasus pada LPK Mulia Mandiri Indonesia (MMI) | Mengatasi permasalahan LPK MMI di masyarakat agar dapat mengetahui informasi pemagangan dengan benar dan mudah secara online.  Untuk menunjang pekerjaan dengan perancangan database dalam pembuatan laporan-laporan yang cepat dan akurat yang bisa disajikan dengan tepat waktu.  Memudahkan calon pemagang untuk melihat persebaran perusahaan apa saja yang tersedia untuk menjadi tempat pemagangan dengan muda sehingga calon pemagang memiliki acuan mengenai perusahaan tempat pemagangan yang akan ditempatinya. | Pada sistem informasi pemagangan LPK Mulia Mandiri Indonesia ini tidak mencakup keberangkatan peserta magang, maka jika ada pengembenagan selanjutnya diharapkan untuk membahas juga mengenai keberangkatan magang.  Pada sistem informasi Pemagangan LPK Mulia Mandiri Indonesia ini tidak menampilkan rekomendasi dan penempatan di perusahaan tempat magang bagi peserta magang, maka diharapkan pada pengembangan selanjutnya untuk membahas mengenai rekomendasi dan penempatan tempat bagi peserta magang  Pembuatan sistem informasi Pemagangan LPK Mulia Mandiri Indonesia ini masih dapat dikembangkan seiring berkembangnya spesifikasi kebutuhan pengguna, terutama dalam segi keamanan website bisa dibuatkan sistem security nya. | Metode penelitian menggunakna model waterfall  Tidak ada penilaian terhadap kinerja mahasiswa nya  Tidak adanya fitur pemberian tugas kepada mahasiswa yang akan magang |
| 5 | Rancang Bangun Sistem Informasi Data Siswa Praktik Kerja Lapangan (PKL) Berbasis Web Responsive pada SMK TI Udayana | Gusti Ayu, Gusti Lanang, I Gede Putu, Jurnal Sistem Informasi STMIK Primakara, 2017 | Dengan memanfaatkan sistem informasi pendataan siswa PKL ini, pihak sekolah dapat secara lebih cepat dan akurat mendapatkan informasi terhadap siswa yang sedang melakukan kegiatan praktek kerja lapangan. Selain itu web ini juga dilengkapi dengan map yang  Memudahkan pembimbing maupun siswa dalam mengetahui alamat perusahaan tersebut. | Dalam sistem ini terdapat informasi data siswa PKL yang dapat diakses oleh user yaitu operator, kepala sekolah, siswa dan guru pembimbing. Sistem ini dapat memudahkan petugas untuk mendata siswa dalam kegiatan PKL dan memudahkan siswa untuk mencari tempat PKL  Pengujian yang telah dilakukan dengan metode blackbox menghasilkan sistem informasi data siswa PKL yang sesuai dengan harapan dan dinyatakan valid  Dari hasil kuesioner efektifitas yang diukur menggunakan perhitungan skala likert maka dapat disimpulkan bahwa sistem informasi data siswa PKL dinyatakan efektif dikembangkan sebagai penunjang kinerja pihak sekolah SMK TI Udayana. | dapat dikembangkan seiring berkembangnya spesifikasi kebutuhan pengguna, terutama dalam segi keamanan website bisa dibuatkan sistem security nya. | Metode penelitian menggunakan model waterfall  Tidak ada penilaian terhadap kinerja mahasiswa nya  Tidak adanya fitur pemberian tugas kepada mahasiswa yang akan magang |
| 6 | Rancang Bangun Sistem Informasi Skripsi dan Kerja Praktik Berbasis *Web* | Adnan Kasim, Lillyan Hadjaratie, Roviana H. Dai, Jurnal Sistem Informasi Universitas Negeri Gorontalo, 2020 | Bertujuan untuk membangun suatu sistem aplikasi yang dapat digunakan untuk mengelola data skripsi dan kerja praktik untuk menunjang kinerja sistem administrasi akademik di Jurusan Teknik Informatika, Universitas Negeri Gorontalo | Sistem aplikasi dapat berjalan dengan baik dalam membantu pengelola dan dosen dalam hal monitoring tahapan, kemajuan bimbingan, perserta ujian seminar, dan riwayat nilai ujian skripsi serta kerja praktik. | Diharapkan untuk menyajikan informasi yang akurat, relevan, dan tepat waktu mengenai periode pendaftaran usulan topik dan judul skripsi, periode pendaftaran ujian | Metode penelitian menggunakan model prototyping.  Terdapat pokok bahasan yang berbeda seperti bimbingan skripsi, ujian seminar, dan ujian skripsi. |