

BAB II

KAJIAN LITERATUR

Pada bab ini akan dijelaskan seluruh landasan teori dan juga penelitian terkait yang dijadikan sebagai acuan dalam menyusun tugas akhir ini. Konsep-konsep yang akan dibahas meliputi :

2.1 Tinjauan Pustaka

2.1.1 Link-Match

Link-match merupakan program pendidikan yang telah didengungkan sejak kabinet pembangunan VI, yang pada dasarnya dilatar belakangi untuk mempertemukan antara dunia pendidikan dengan dunia ketanaga kerjaan (Potiua, 2003).

Pengertian *Link-match* secara harafiyah adalah *link* yang berarti bertautan, keterkaitan atau berhubungan sedangkan *match* berarti kecocokan atau kesesuaian. Dengan demikian dapat disimpulkan bahwa konsep *Link-match* adalah meneken dunia pendidikan untuk menghasilkan produk pendidikan yang dapat menangani masalah-masalah yang sifatnya praktis-teknis dan berjangka pendek (Potiua, 2003). Dalam artian lain *Link-match* merupakan sebuah metode belajar yang merefleksikan terminologi yang bersumber dari aliran belajar *behavior* atau yang dikenal juga dengan *learning by doing* (*Program Class Link & Match di STT Terpadu Nurul Fikri*, 2019).

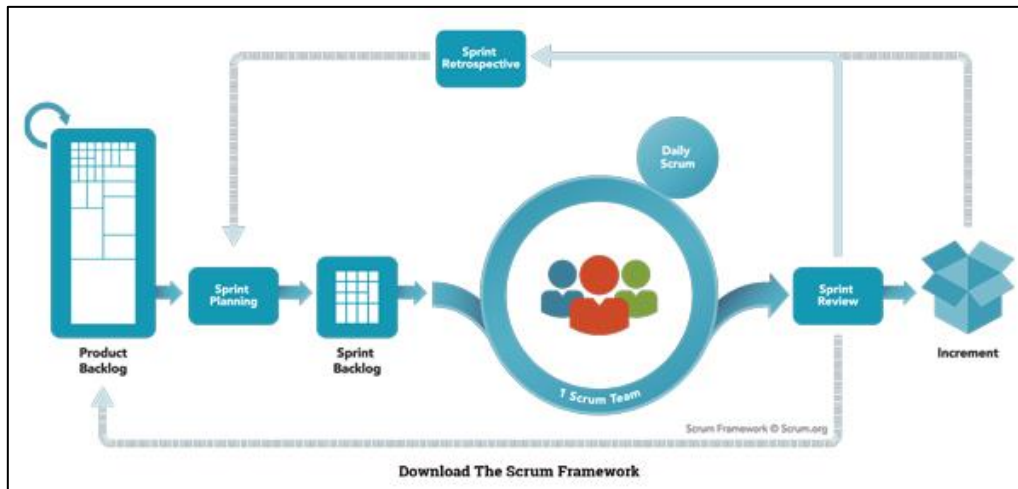
Pada tahun 2019, STT Terpadu Nurul Fikri membuat metode pembelajaran *Link-match* dan metode pembelajaran ini menggabungkan beberapa mata kuliah yang saling berkaitan dalam pembuatan sebuah proyek pembuatan perangkat lunak. Pada metode *Link-match*, akan memberikan implementasi kepada mahasiswa tentang bekerja dalam sebuah tim pengembangan perangkat lunak dan untuk memenuhi kebutuhan industri kedepannya.

2.1.2 Model Pengembangan

2.1.2.1 SCRUM

SCRUM adalah kerangka kerja proses yang telah digunakan untuk mengelola pengembangan produk kompleks yang dalam pengerjaannya dapat menggunakan bermacam proses dan teknik. *SCRUM* berfungsi untuk meperjelas ketidak-efektifan dari manajemen

produk dan teknik kerja, sehingga kinerja produk, tim, dan lingkungan kerja dapat terus ditingkatkan.



Gambar 2. 1 SCRUM (The SCRUM Framework Poster | SCRUM.org, n.d.)

Kerangka kerja *SCRUM* terdiri dari *SCRUM Team* dan peran-peran, acara-acara, artefak-artefak dan aturan terkait. Setiap komponen di dalam kerangka kerja ini memiliki tujuan tertentu dan sangat penting bagi keberhasilan penggunaan *SCRUM*. Peran-peran, acara-acara dan artefak-artefak terikat dengan aturan *SCRUM* untuk menjaga hubungan dan interaksi antar komponen tersebut. (Schwaber & Sutherland, 2017)

a. *SCRUM Team*

SCRUM Team bersifat swakelola yang dimana tim dapat memilih cara terbaik dalam mengerjakan pekerjaan mereka. Selain itu *SCRUM Team* juga bersifat multi-fungsi yang dimana tim memiliki semua keahlian yang dibutuhkan untuk menyelesaikan pekerjaan mereka tanpa bergantung dengan orang lain. (Wijaya, 2018) *SCRUM Team* Terdiri dari:

- ***Product Owner***

Product owner adalah seorang profesional yang memiliki pengetahuan yang mendalam mengenai produk / proyek yang akan dikerjakan.

- ***Development Team***

Development Team merupakan tim yang beranggotakan individu-individu yang mampu mengorganisir dirinya sendiri dan merupakan sebuah tim yang multi-fungsi.

- ***SCRUM master***

SCRUM master adalah orang yang bertanggung jawab untuk mengarahkan tim *SCRUM / development team* supaya bekerja menuju arah dan tujuan yang benar dan mengatasi semua rintangan dan halan yang dapat mempengaruhi kinerja mereka.

b. *SCRUM Events*

Event dalam *SCRUM* diselenggarakan guna terciptanya kerutinan dan mengurangi pertemuan lain yang bukan merupakan bagian dari *SCRUM* (Schwaber & Sutherland, 2017). Berikut *event* yang ada dalam *SCRUM*:

- ***Sprint***

Sprint adalah inti dari kegiatan *SCRUM* ini, dimana *Sprint* sebagai sebuah batasan waktu dalam pengerjaan proyek dalam *SCRUM*.

- ***Sprint Planning***

Dalam *Sprint planning* ini semua pekerjaan yang akan dikerjakan dalam *Sprint* dirancang. Dan dalam *Sprint planning* ini *goal* harus ditentukan sebagai titik akhir dari proyek yang dikerjakan.

- ***Daily SCRUM***

Daily SCRUM berfungsi untuk menginspeksi perkembangan pekerjaan menuju *Sprint goal* dan tren perkembangan penyelesaian pekerjaan di *Sprint Backlog*. Dengan adanya *daily SCRUM* ini kemungkinan untuk tim mencapai *Sprint goal* dapat meningkat.

- ***Sprint Review***

Sprint reView diselenggarakan di akhir *Sprint* untuk menginspeksi hasil satu *Sprint* yang telah dilaksanakan, hal ini dilakukan guna mengoptimalkan nilai bisnis dari proyek tersebut.

- ***Sprint Retrospective***

Sprint retrospective adalah sebuah kesempatan bagi *SCRUM team* untuk mengevaluasi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

c. **SCRUM Artifacts**

SCRUM artifacts merepresentasikan pekerjaan atau nilai bisnis guna terciptanya transparansi dan kesempatan untuk menginspeksi dan mengadaptasi. Artefak-arte-fak yang dijabarkan *SCRUM* dirancang sedemikian rupa untuk memaksimalkan transparansi informasi utama agar setiap orang memiliki pemahaman yang sama mengenai artefak tersebut (Schwaber & Sutherland, 2017).

- **Product Backlog**

Product backlog adalah daftar yang berisi semua hal yang perlu diketahui dan harus ada dalam produk. *Product backlog* adalah satu-satunya sumber kebutuhan untuk semua perubahan yang perlu diberlakukan terhadap produk.

- **Sprint Backlog**

Sprint backlog adalah daftar *product backlog* yang terpilih untuk dikerjakan dalam *Sprint* untuk menghantarkan *increment* dan mencapai *Sprint goal*.

2.1.2.2 **Unified Modeling Language (UML)**

Unified Modeling Language (UML) adalah sebuah metode yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Dharwiyanti & Wahono, 2003). UML merupakan metodologi dalam pengembangan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

- a. **Use Case Diagram**

(Hendini, 2016) *Use Case Diagram* adalah bentuk peModelan untuk menggambarkan apa yang akan dilakukan oleh sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sistem dan siapa saja yang berhak menggunakan fungsi tersebut.

- b. **Activity Diagram**

(Hendini, 2016) *Activity Diagram* digunakan untuk menggambarkan alur kerja atau aktivitas dari sistem yang dibangun.

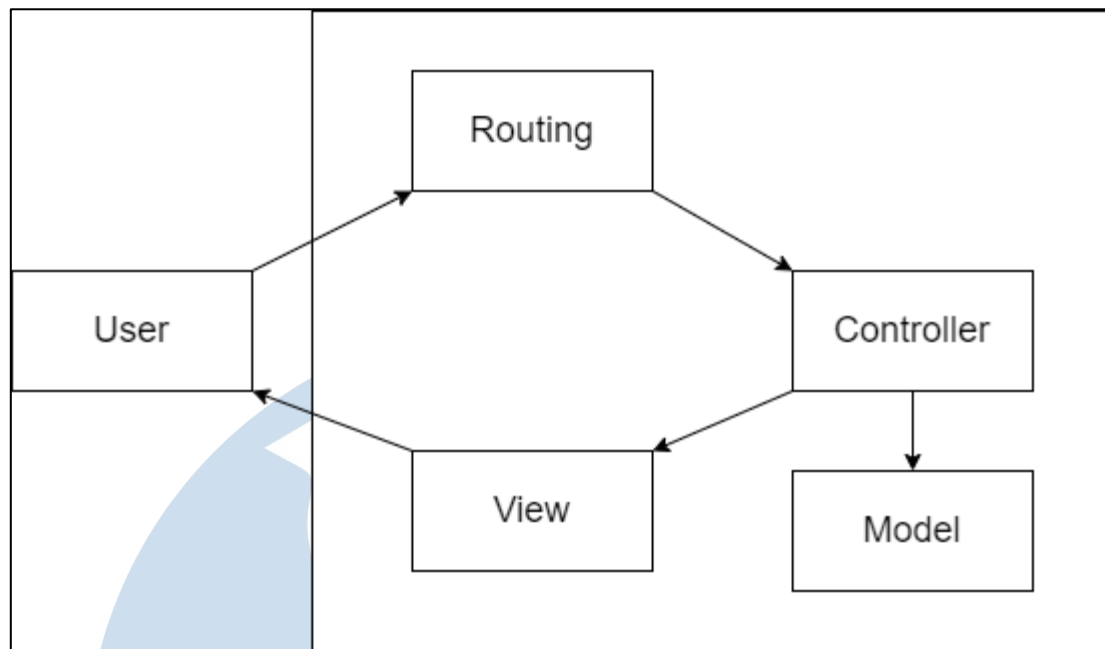
2.1.3 Tools Pengembangan

2.1.3.1 PHP Framework Laravel

Framework adalah sebuah *software* yang berisi berbagai fungsi, plugin, dan konsep sehingga membentuk suatu sistem tertentu, dengan begitu penggunaan *Framework* dapat mempermudah programmer dalam membuat aplikasi atau *web*. Selain itu dengan menggunakan *Framework*, sebuah aplikasi akan tersusun dan terstruktur rapi (*Pengenalan Framework PHP*, n.d.). Dalam pembangunan sebuah aplikasi aplikasi yang elegan dan dinamis, laravel adalah salah satu *Framework* yang menjadi primadona bagi para programmer PHP, karna *Framework* ini menekankan kesederhanaan dan fleksibilitas pada desainnya (Aminudin, 2015). Sama seperti *Framework* lainnya, laravel dibangun dengan konsep MVC (*Model-Controller-View*), kemudian laravel dilengkapi dengan *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan instalasi bundle melalui *command prompt*. Selain itu laravel juga memiliki banyak kelebihan, diantaranya: a) *Expressif*; b) Sederhana karna ada *Eloquent ORM*; dan c) *Accessible*. Kelebihan-kelebihan tersebut didukung dengan banyaknya fitur yang dimiliki oleh laravel, diantaranya:

- a. *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular
- b. *Eloquent ORM*, merupakan penerapan PHP lanjutan dari pola “*active record*” menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*.
- c. *Application Logic*, merupakan aplikasi yang dikembangkan menggunakan *Controllers* maupun bagian dari deklarasi *Route*.
- d. *Reverse Routing*, untuk mendefinisikan hubungan antara *Link* dan *Route*.
- e. *Resful Controllers*, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP *GET* dan permintaan *POST*.
- f. *Class Auto Loading*, menyediakan *loading* otomatis untuk *class-class* PHP.
- g. *View Composers*, adalah kode unit *logical* yang dijalankan ketika sebuah *View* di *load*.
- h. *IoC Container*, objek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik
- i. *Migrations*, sistem *control* untuk skema *database*.
- j. *Unit Testing*, untuk mendeteksi dan mencegah regresi.
- k. *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman dengan metode otomatis yang terintegrasi ke laravel.

2.1.3.2 MVC



Gambar 2. 2 MVC

MVC adalah singkatan dari *Model View Controller*. MVC merupakan teknik pemrograman yang membagi aplikasi menjadi tiga bagian, yaitu *Model* yang berisi fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan *database*, *View* yang berisi fungsi kode untuk memberikan sebuah tampilan pada aplikasi yang memberikan informasi kepada pengguna, dan *Controller* sebagai perantara antara *Model* dan *View* dan semua sumber yang dibutuhkan untuk memproses permintaan pada aplikasi (Prabowo, 2015).

Pada *Framework* Laravel terdapat satu struktur tambahan selain *Model*, *View*, dan *Controller* yaitu *Routing*. *Routing* pada *Framework* Laravel berfungsi untuk menangani *request* yang diberikan *user* kepada aplikasi *web*. Dengan kata lain *Routing* akan menjadi jembatan yang menghubungkan *user* dengan aplikasi *web*, sehingga setiap *request* pada alamat tertentu akan dieksekusi terlebih dahulu sebelum menampilkan *response*. (Nugraha, 2014)

2.1.3.3 PostgreSQL

PostgreSQL adalah *database* yang dikembangkan oleh *University of California* di *Berkeley Computer Science Department*, yang menjadi salah satu alternatif solusi bagi pengguna *database* yang mendukung banyak *platform* dan bebas lisensi. *PostgreSQL* juga merupakan *database* yang bersifat *open source* yang menjadikannya dapat dikembangkan sesuai dengan kebutuhan (Munawaroh, 2005).

PostgreSQL juga termasuk kedalam ORDBMS (*Object Relational Database Management System*) sehingga membuat *PostgreSQL* memiliki berbagai kemampuan yang dimiliki oleh *database* komersil umum lainnya, seperti dukungan menggunakan perintah-perintah SQL memungkinkan Administrator lebih mudah berinteraksi dengan *database PostgreSQL*. Dengan kemampuan untuk dapat mevariasikan perintah *select* dengan berbagai macam klausa yang ada, menjadikan *select* pada *database* ini jauh lebih fleksibel dalam melakukan *query* data dari tabel-tabel yang ada. selanjutnya kemampuan dalam hal manajemen *user* yang dapat mengakses *database*, menjadikan keamanan *PostgreSQL* lebih terjamin (Munawaroh, 2005).

PostgreSQL juga dapat membuat *performance* dari aplikasi yang dibuat dapat lebih optimal, karena dengan adanya kemampuan untuk membuat *function*, *stored procedure* dan *trigger* membuat beberapa proses yang seharusnya diproses di komputer *client* dapat di proses di komputer sever. Dengan begitu aplikasi yang berjalan di komputer *client* benar-benar aplikasi *thin client*. Keunggulan lainnya dari *database* ini yaitu, *PostgreSQL* mendukung sebagai media penyimpanan dari berbagai bahasa pemrograman seperti : Java, Python, PHP, Perl, dan lain-lain. (Munawaroh, 2005).

2.1.4 Pengujian Sistem

2.1.4.1 Black box Testing

Black box adalah salah satu teknik pengujian perangkat lunak yang dimana pengujian diambil dari sudut pandang pengguna, sehingga dalam pengujian para tester akan memandang perangkat lunak layaknya sebuah “kotak hitam” yang artinya dalam pengujiannya para tester tidak perlu mengetahui seperti apa kinerja internal dan struktur program dari perangkat lunak tersebut (Anisah et al., 2016). Pengujian *black box* berusaha untuk menemukan kesalahan dalam kategori (Setiawan, 2011):

- a. Fungsi-fungsi yang tidak benar atau hilang,
- b. Kesalahan *interface*,
- c. Kesalahan dalam struktur data atau akses *database* eksternal,
- d. Kesalah kinerja,
- e. Inisialisasi dan kesalahan terminasi.

2.1.4.2 User Acceptance Testing (UAT)

UAT adalah sebuah proses pengujian yang melibatkan pengguna langsung untuk menghasilkan dokumen yang menunjukkan bahwa aplikasi yang dikembangkan sudah sesuai dengan kebutuhan dan dapat diterima oleh pengguna (Supriatna, 2018). Tahap pengujian sendiri dilakukan dengan menggunakan *in-depth interView*, dimana pewawancara dan responden akan melakukan tanya jawab untuk mendapatkan hasil dari pengujian yang dilakukan (Wang et al., 2019).

2.1.4.3 Kuesioner

Kuesioner merupakan metode pengumpulan data yang efisien karena berisi daftar pertanyaan yang tertulis dan diajukan kepada responden, setiap jawaban yang diberikan dari semua pertanyaan dalam kuesioner dicatat dan direkam. Akan tetapi peneliti harus mengetahui dengan pasti data atau informasi yang dibutuhkan dan bagaimana variabel yang menyatakan informasi yang dibutuhkan tersebut. Dan pertanyaan yang diberikan haruslah jelas dan mudah dimengerti untuk mengurangi kesalahan interpretasi responden dalam pengisian kuesioner (Widi, 2011).

2.1.4.4 Skala Likert

Skala *likert* adalah skala pengukuran yang dikembangkan oleh *Likert* pada tahun 1932. Skala *likert* mempunyai empat atau lebih butir-butir pernyataan yang dikombinasikan sehingga membentuk sebuah nilai yang mempresentasikan sifat individu (Budiaji, 2013). Butir-butir pernyataan tersebut terdiri dari. Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS) baik pernyataan yang positif maupun pernyataan yang negatif (Intang Sappaile, 2007).

2.2 Penelitian Terkait

2.2.1 Tabel Penelitian

Pada bagian penelitian terkait, penulis melakukan studi literatur penelitian terkait berdasarkan permasalahan yang peneliti ambil. Penulis menganalisis penelitian terkait untuk mengetahui tingkat kebutuhan permasalahan yang sedang diteliti oleh penulis. Penelitian terkait juga bertujuan untuk mengetahui posisi penelitian yang sedang dilakukan. Berikut daftar penelitian terkait dapat dilihat pada tabel berikut ini:

Tabel 2. 1 Penelitian Terkait

NO	Nama dan Tahun	Judul	Topik	Tools	Metodologi	Hasil
1.	Heni Jusuf, Ariana Azimah, Winarsih Univesitas Nasional 2013	Pengembangan Aplikasi Absensi Dosen dengan Menggunakan <i>Fingerprint</i> (Sidik Jari Digital) di Univesitas Nasional	Membangun sistem absensi dosen menggunakan <i>fingerprint scanner</i>	<i>Fingerprint Scannser,</i>	<i>Prototyping</i>	Membuat aplikasi absensi menggunakan <i>fingerprint</i> , yang dapat membantu SPA dalam memonitoring kehadiran dosen dan meminimalisir terjadinya kesalahan
2.	Noval Aditya Muhammad, Febriliyan Samopa, Radityo Prasetyanto Wibowo Institut Teknologi Sepuluh Nopember Surabaya 2013	Pembuatan Aplikasi Presensi Perkuliahan Berbasis <i>Fingerprint</i> (Studi kasus : Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya)	Membangun aplikasi untuk mengelola data presensi <i>fingerprint</i> guna menanggulangi kecurangan presensi	<i>Fingerprint Scanner,</i> CodeIgniter (CI), PHP	<i>Web-Based Application</i>	Membuat aplikasi yang dapat melakukan monitoring absensi <i>fingerprint</i> , sehingga mempermudah Admin /sekretariat dalam merekapitulasi absensi

NO	Nama dan Tahun	Judul	Topik	Tools	Metodologi	Hasil
3.	Gat STMIK Pontianak 2016	Integrasi <i>Fingerprint</i> System Dengan Real Time Absensi Dosen Berbasis <i>Web</i> (Studi Kasus : STMIK Pontianak)	Membangun Sistem Absensi dosen menggunakan <i>Fingerprint</i> Secara <i>Real time</i> berbasis <i>Web</i>	<i>Fingerprint Scanner</i> , <i>Unified Modeling</i> Sistem (UML)	<i>Research and Development</i> (R&D), <i>Rapid Application Development</i> (RAD)	Membuat sistem absensi <i>fingerprnt</i> secara <i>realtime</i> , sehingga dapat mempermudah pimpinan dalam memonitoring prsensi dosen.
4.	Elisabeth Febrina Tuto Burak Lamatokan Universitas Sanata Dharma Yogyakarta 2016	Sistem Presensi Mahasiswa Berabasis <i>Web</i> Menggunakan <i>Finger Print Scanner</i> (Studi Kasus program studi Teknik Informatika Universitas Sanata Dharma Yogyakarta)	Membangun sistem presensi mahasiswa berbasis <i>web</i> menggunakan identitas sidik jari	<i>Fingerprint Scanner</i> , PHP, <i>Database</i> MySql	FAST (<i>Framework for the Application of Systems Technique</i>)	Membuat aplikasi kehadiran berbasis <i>web</i> dan menggunakan <i>fingerprint</i> scanner, yang dapat membantu <i>Admin</i> /sekretariat dalam mengolah data kehadiran, dan dapat membantu mahasiswa dalam mengetahui status serta persentase kehadirannya.

STT - NF

2.2.2 Posisi Penelitian

Tabel 2. 2 Posisi Penelitian

No	Kehadiran	Berbasis Web	Laravel	SCRUM
1.	<u>Heni Jusuf, Ariana Azimah, Winarsih (2013)</u> Pengembangan Aplikasi Absensi Dosen dengan Menggunakan <i>Fingerprint</i> (Sidik Jari Digital) di Univesitas Nasional			
2.	<u>Gat (2016)</u> Sistem Presensi Mahasiswa Berabasis <i>Web</i> Menggunakan Finger Print Scanner (Studi Kasus program studi Teknik Informatika Universitas Sanata Dharma Yogyakarta)			
3.	<u>Noval Aditya Muhammad, Febriliyan Samopa, Radityo Prasetianto Wibowo (2013)</u> Pembuatan Aplikasi Presensi Perkuliahan Berbasis <i>Fingerprint</i> (Studi kasus : Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya)			
4.	<u>Elisabeth Febrina Tuto Burak Lamatoka (2016)</u> Sistem Presensi Mahasiswa Berabasis <i>Web</i> Menggunakan Finger Print Scanner (Studi Kasus program studi Teknik Informatika Universitas Sanata Dharma Yogyakarta)			
5.	<u>Ahmad Fathan Syakir (2020)</u> Rancang Bangun Aplikasi Link – Match STT-NF Modul Kehadiran Berbasis <i>Framework</i> Laravel			