

## BAB II

### KAJIAN LITERATUR

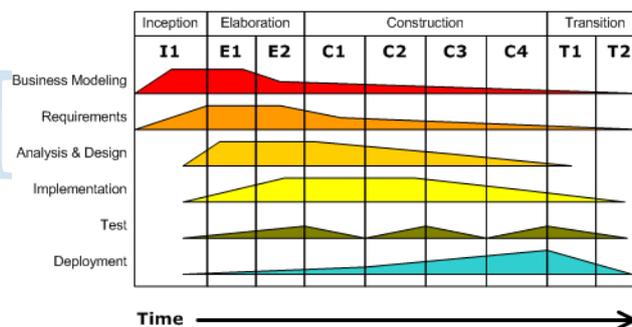
Dalam bab ini akan dijelaskan mengenai teori, algoritma, *model*, fokus teknologi yang digunakan, teori terkait dan relevansi dengan judul tugas akhir.

#### 2.1 Tinjauan Pustaka

##### 2.1.1 *Unified Process*

Unified Process merupakan salah satu *software development proses* yang telah mendukung konsep berorientasi objek, Unified Process menggunakan *Unified Modeling Language (UML)* sebagai notasi dalam menggambarkan proses pengembangan perangkat lunak. Dalam *Unified Process*, pengembangan perangkat lunak dilakukan dalam sejumlah siklus yang berkelanjutan. Setiap siklus akan berakhir pada *release* sebuah produk. Setiap siklus tersebut terdiri atas empat fase yang berbeda, yaitu *inception*, *elaboration*, *construction*, dan *transition*. Setiap fase dapat terdiri atas sejumlah iterasi *workflow (requirement, analysis, design, implementation, dan test)*. Selanjutnya, *workflow* menentukan urutan aktivitas yang perlu dilakukan dan artifak apa yang harus dihasilkan [3].

**Iterative Development**  
Business value is delivered incrementally in time-boxed cross-discipline iterations.



Gambar 2.1 Contoh Unified Process

### 2.1.2 *Repository*

Pendapat lain menurut (Pfister, 2008), menyatakan bahwa perguruan tinggi yang berbasis *repository* adalah satu set layanan yang menawarkan berbagai bahan digital yang dihasilkan oleh lembaga tersebut ataupun yang dihasilkan lembaga lain yang dikelolanya kepada masyarakat penggunanya. Clifford Lynch (2003), mendefinisikan *repository* pada perguruan tinggi adalah serangkaian pelayanan yang diberikan oleh perguruan tinggi kepada mahasiswa untuk mengelola dan menyebarluaskan bahan-bahan digital yang dihasilkan oleh perguruan tinggi. Bahan-bahan digital yang dimaksud adalah seluruh karya ilmiah dan/atau output intelektual yang dihasilkan oleh suatu perguruan tinggi. Ada juga yang mendefinisikan *repository internal* adalah tempat menyimpan seluruh karya yang dihasilkan oleh civitas akademika suatu perguruan tinggi dan/atau karya lain mengenai perguruan tinggi yang bersangkutan [4].

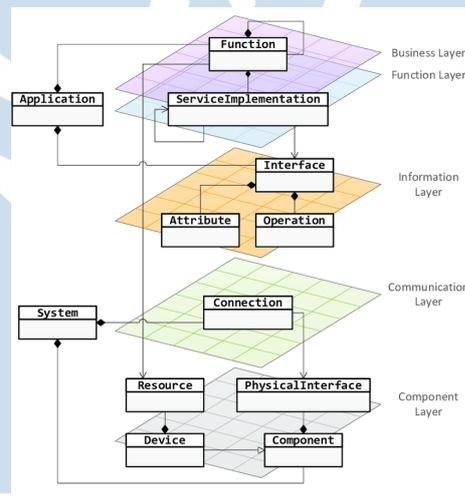
### 2.1.3 UML (Unified Modeling Language)

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifak (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifak tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti dalam pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep berorientasi objek. UML dibuat oleh Grady Booch, James Rumbaugh dan Ivar Jacobson dibawah bendera *Rational Software Corp.* UML menyediakan notasi yang membantu memodelkan perangkat lunak, namun hampir dalam sebuah bidang yang membutuhkan pemodelan. [5].

UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk mendokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek. UML sudah menjadi *standardized modelling language* yang terdiri dari kumpulankumpulan diagram,

dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti .

1. Spesifikasi
2. Visualisasi
3. Desain Arsitektur
4. Kontruksi
5. Simulasi dan *Testing*
6. Dokumentasi
7. Sketsa

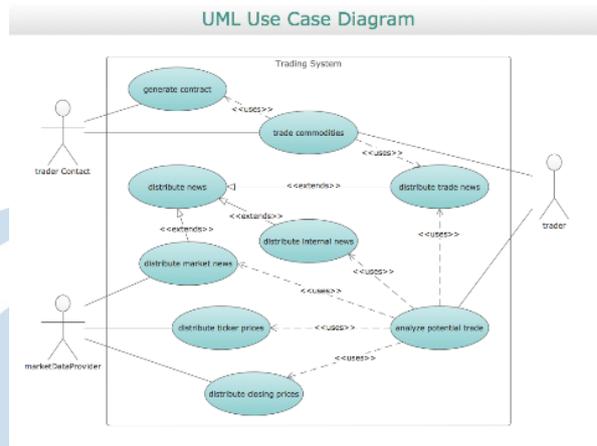


Gambar 2.2 Contoh Unified Modeling Language

Didalam UML terdapat pengelompokan diagram sistem. Diagram ini berbentuk grafik yang menunjukkan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

Jenis-jenis diagram dalam UML [4].

a. *Use case Diagram*

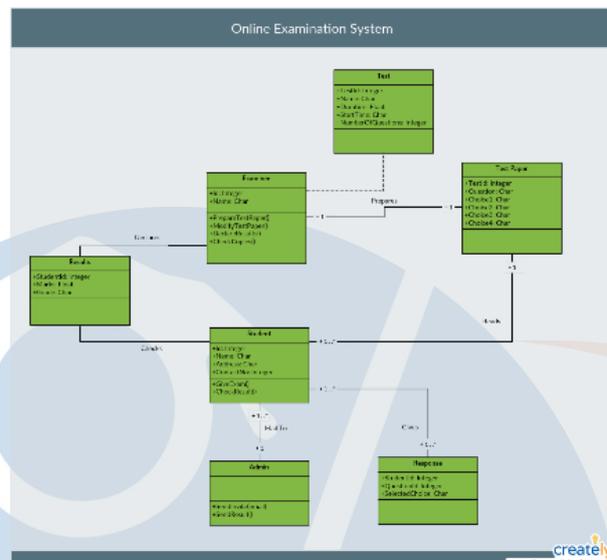


Gambar 2.3 Contoh usecase

Menggambarkan sejumlah eksternal *actors* dan hubungannya ke *Use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dari dokumentasi dari *Use case* symbol namun juga dapat dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

STT - NF

b. *Class Diagram*



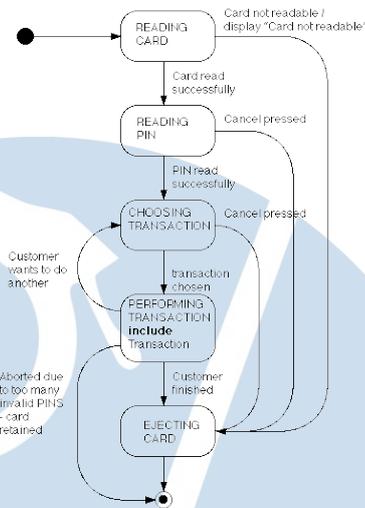
Gambar 2.4 contoh class diagram

Menggambarkan struktur *static class* di dalam sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara : *associated* (terhubung satu sama lain), *depended* (satu *class* tergantung atau menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* yang lainnya), atau *packed* (group bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

STT - NF

c. *State Diagram*

State-Chart for One Session

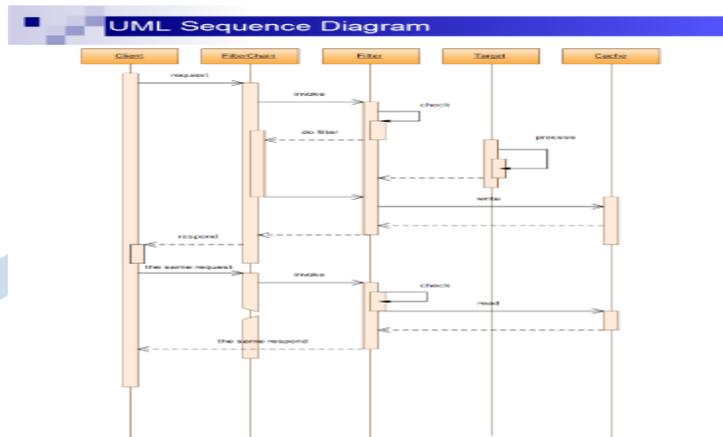


Gambar 2.5 contoh State digram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu objek dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa objek lain yang mengirim pesan, state class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

STT - NF

#### d. *Sequence Diagram*



Gambar 2.6 contoh *Sequence diagram*

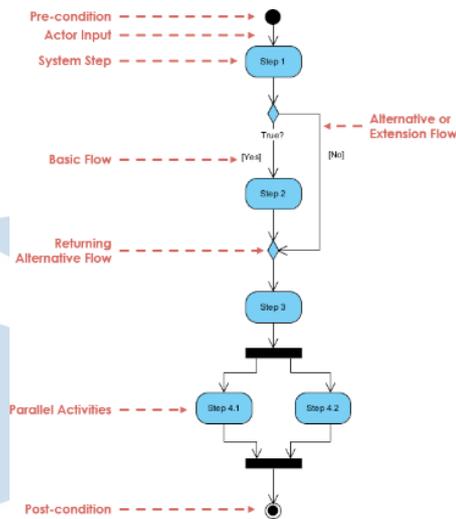
Menggambarkan kolaborasi dinamis antara sejumlah objek. Kegunaannya untuk menunjukkan sejumlah pesan yang dikirim antara objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

#### e. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan, gunakan *sequence diagram* tapi jika penekanannya pada konteks gunakan *collaboration diagrams*.

STT - NF

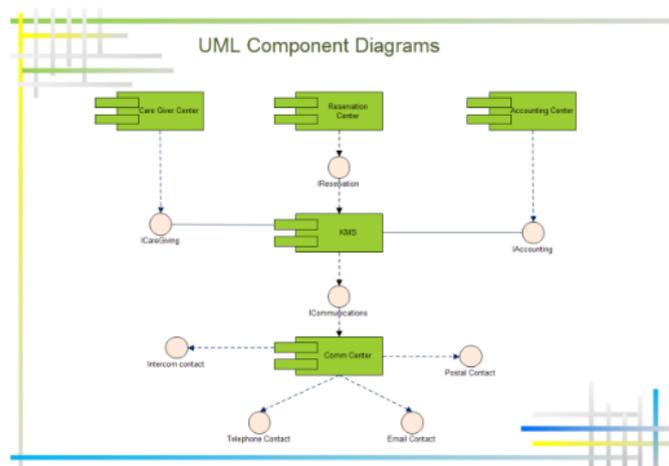
f. *Activity Diagram*



Gambar 2.7 contoh Activity diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *Use case* atau interaksi.

g. *Component Diagram*

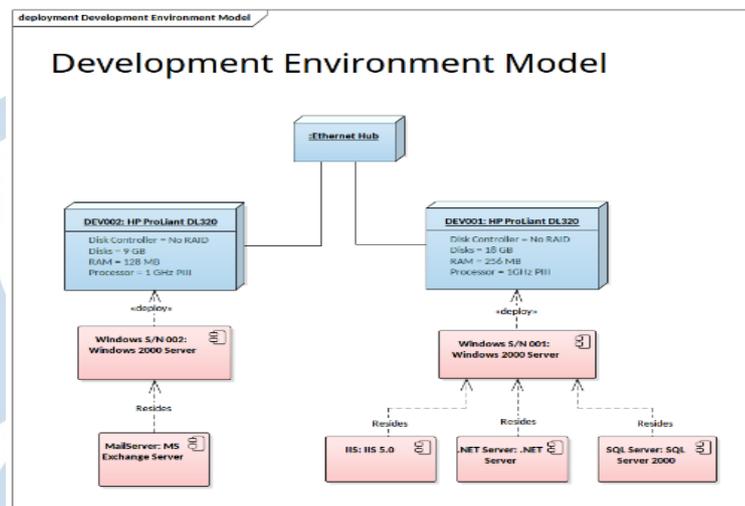


Gambar 2.8 Contoh Component diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa source code, komponen biner atau *class* yang

diimplementasikan sehingga membuat pemetaan dari logical view ke component view.

#### h. Deploy Diagram



Gambar 2.9 Contoh Deploy Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, executable dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

#### 2.1.4 Framework

*Framework* adalah kerangka kerja. Framework digunakan oleh *developer* untuk memudahkan pembangunan aplikasi *web* yang dapat berupa sekumpulan library yang berisi fungsi, *tools*, ataupun *class-class*, dan digunakan sebagai kerangka dalam pembangunan aplikasi *web*. Umumnya didalam framework tersebut telah menyediakan solusi untuk akses database, *authentication*, *templating*, *controls*, dan fungsi-fungsi lainnya. Penggunaan framework diharapkan membuat pengembangan

aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan *reusable*. [6].

### 2.1.5 Yii Framework

Yii adalah kerangka kerja PHP berbasis-komponen dengan kinerja tinggi untuk pengembangan aplikasi *web* berskala-besar. Ia menyediakan *reusability* maksimum dalam pemrograman *web* dan bisa mengakselerasi proses pengembangan secara signifikan. Nama Yii (dieja sebagai /i:/) singkatan dari "*Yes It Is!*" *easy, efficient* dan *extensible* (mudah, efisien, dan bisa diperluas). Yii adalah kerangka kerja pemrograman umum *web* yang bisa dipakai untuk mengembangkan semua jenis aplikasi *web*. Oleh karena Yii sangat ringan dan dilengkapi dengan solusi caching yang memuaskan, ia sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (CMS), sistem e-commerce, dll. Seperti kebanyakan PHP framework, ensi, kaya-fitur, dan didokumentasikan dengan jelas. Yii adalah hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi *web* dan investigasi serta refleksi kerangka kerja pemrograman *web* paling populer dan aplikasi. Yii saat ini mempunyai 2 versi utama: 1.1 dan 2.0. Versi 1.0 adalah generasi lama dan sekarang dalam mode pemeliharaan. Versi 2.0 adalah penulisan ulang lengkap dari Yii. Mengadopsi teknologi terbaru dan protokol, termasuk composer, PSR, namespaces, trait, dan sebagainya. Versi 2.0 merupakan generasi sekarang dari framework Yii [2].

### 2.1.6 Black Box Testing

Black box testing dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai behavioral testing, specification-based testing, input/output testing atau functional testing. Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software.

Black-box testing merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan kotak hitam yang perilakunya hanya dapat ditentukan dengan mempelajari input dan output yang berkaitan. Nama lain untuk pengujian ini ialah pengujian fungsional karena penguji hanya berkepentingan dengan fungsionalitas dan bukan implementasi perangkat lunak [7].

### **2.1.7 User Acceptance Testing (UAT)**

Pada penelitian ini penulis menggunakan *User Acceptance Testing* atau biasa disingkat dengan UAT. User Acceptance Testing (UAT) merupakan salah satu metodologi yang sangat inovatif untuk mencegah kegagalan proyek IT [8]. Dalam pengembangan perangkat lunak, terdapat tiga hal yang dilakukan dalam proses UAT yaitu [9]:

- a. UAT mengukur bagaimana sistem sudah sesuai dengan kebutuhan pengguna
- b. UAT mengekspos fungsionalitas/logic bisnis yang belum ditemukan, karena unit testing dan sistem testing tidak berfokus pada fungsionalitas/ logic bisnis.
- c. UAT membatasi bagaimana sistem telah “selesai” dibuat.

Pengujian kali ini dilakukan terhadap calon pengguna dengan jumlah sebanyak 4 responden dari Mahasiswa. Diharapkan penelitian ini akan menjadi bermanfaat kepada mahasiswa STT Terpadu Nurul Fikri.

## **2.2 Penelitian terkait**

Dalam penelitian ini melakukan studi literatur penelitian terkait sebagai komparasi dan berkaitan dengan masalah yang penelitian ambil. Hal ini bertujuan untuk mengetahui posisi penelitian yang dilakukan peneliti. Daftar penelitian terkait yang peneliti temukan bisa dilihat di label 2.1.

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Studi Kasus	Tools	Teknologi	Kesimpulan	Saran
1	Ninda Nusliyani, Ramos Somya,S.Kom.,M.Cs, Christine Dewi ,S.Kom.,M.Cs, 2016	Penerapan Yii <i>Framwork</i> pada Pengelolaan Data Proposal FTI UKSW	Universitas Kristen Satya Wacana	Yii	<i>web</i>	Hasil pengujian manfaat yang dilakukan juga menunjukkan jika sistem informasi ini bermanfaat bagi bidang TA dalam melakukan pendataan pendaftaran proposal mahasiswa skripsi dalam aplikasi yang utuh.	(1) Perlu adanya penambahan fitur ekspor ke XML untuk <i>record</i> data (2) Perlu dibuat solusi pencarian data yang lebih efisien tidak berbasis query SQL ke basis data namun bias menggunakan Teknik data mining atau algoritma pencarian lain.
2	Antonius Rachmat C, 2014	Analisis Rancangan Bangun Sistem	Universitas Kristen Duta	Dublin Core	<i>web</i>	Hasil pengujian manfaat yang dilakukan juga	Library yang dirasa manfaatnya pada

		<i>Repository</i> Instuti Berbasis Metadata Dublin Core di UKDW Yogyakarta	Wacana Yogyakarta			menunjukkan jika sistem informasi ini bermanfaat bagi bidang TA dalam melakukan pendataan pendaftaran proposal mahasiswa skripsi dalam aplikasi yang utuh.	penelitian ini adalah library editable kartik dan library moonland php excel , di mana bidang TA tidak akan kerepotan dalam hal sortir proposal per progdi dan dalam hal update data bidang TA dapat mengedit data secara live dalam kolom tersebut sehingga aplikasi tersebut berjalan secara utuh.
3	Sriati Suwanto, 2016	Manajemen Layanan <i>Repository</i> Perguruan Tinggi	Perguruan Tinggi	(tidak ada) dikarenakan penelitian tersebut	(tidakada) Penulis Tidak	Kekuatan utama Repositori Institusi terletak pada informasi muatan lokal yang	hal yang perlu dipersiapkandalam pengembangan manajemen repositori

				seperti studi literatur	membuat sistem	diunggah dan diterbitkan ke dalam repositori, karena informasi muatan lokal sifatnya khas, khusus hanya dimiliki oleh institusi yang mengembangkan repositori.	perguruan tinggi yang perlu dipersiapkan dalam manajemen IR yaitu: Benchmarking adalah SDM, perangkat keras dan lunak, prosedur dan dukungan pimpinan dan manajemen informasi muatan lokal.
4	Saefulloh,2016	Analisis Perancangan Aplikasi Manajemen <i>Repository</i> Karya Ilmiah Berbasis <i>web</i> Menggunakan YII <i>Framework</i>	STT Nurul Fikri	Yii	<i>web</i>	(1) Pengelolaan karya ilmiah menjadi lebih teratur, rapi dan terklarifikasi (2) Aplikasi <i>repository</i> karya ilmiah dapat diakses via jaringan internet	Pada penelitian ini penulis melakukan proses analisis terhadap pengelolaan <i>repository</i> karya ilmiah yang sesuai bidang peminatan yaitu teknologi

menggunakan  
*browser*

perangkat lunak,  
infrastruktur TI, tata  
Kelola SI dan  
Enterprise SI dan  
manajemen user tetapi  
pada proses  
manajemen user  
belum dapat penulis  
implementasikan di  
karenakan faktor  
waktu yang singkat,  
diharapkan untuk  
pengembangan  
selanjutnya proses  
manajemen user dapat  
dilengkapi

STT - NF

Penelitian ini memiliki judul Perancangan Sistem *Repository* Tugas STT Terpadu Nurul Fikri Berbasis *web* dengan menggunakan *tools* Yii dan memanfaatkan teknologi *web*. Penelitian menggunakan lokasi studi kasus di STT Terpadu Nurul Fikri. Penelitian ini akan mengembangkan sistem *repository* untuk tugas kuliah yang akan diunggah oleh mahasiswa STT -NF.



STT - NF