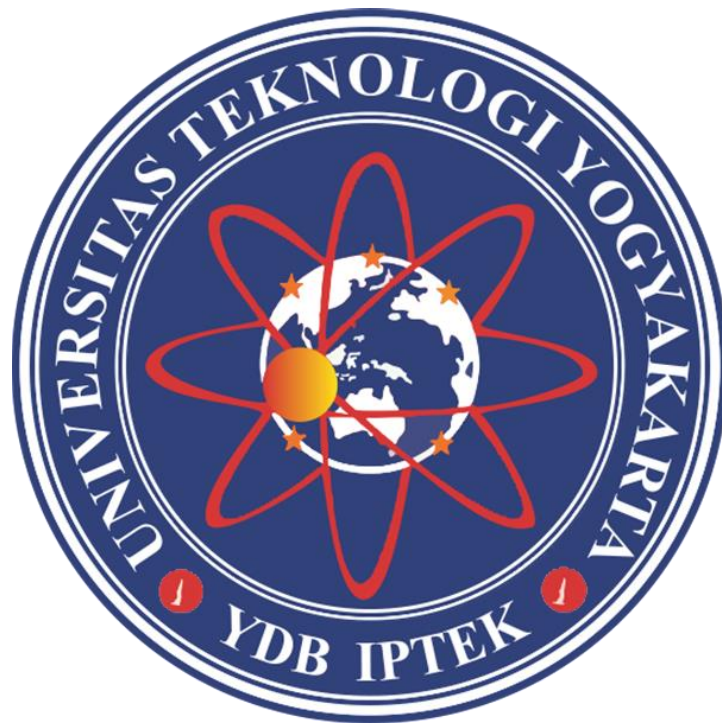


**METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK XII**  
**LAPORAN METODE AGILE UNIFIED PROCESS (AUP)**



**Oleh:**

**5200411007 - ADI KANNATASIK**  
**5200411010 - FARIZ YUDO PRASETYO**  
**5200411012 - MARTIN SETYAWAN WIBOWO**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO**  
**2021/2022**

# Daftar Isi

<b>METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK XII .....</b>	<b>1</b>
<b>BAB 1 .....</b>	<b>3</b>
<b>PENDAHULUAN.....</b>	<b>3</b>
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
<b>BAB 2.....</b>	<b>4</b>
<b>PEMBAHASAN.....</b>	<b>4</b>
2.1 Pengertian.....	4
2.2 Tujuan .....	4
2.3 Manfaat.....	4
2.4 Kelebihan dan Kekurangan .....	5
1. Kelebihan AUP.....	5
2. Kekurangan AUP .....	5
<b>BAB 3.....</b>	<b>6</b>
<b>METODE PENGEMBANGAN SISTEM .....</b>	<b>6</b>
<b>ANALISA DAN PEMBAHASAN .....</b>	<b>7</b>
A. Perancangan .....	7
B. Perancangan Sistem Usulan (Sistem Informasi e-Procurement) .....	7
UC Diagram Pengadaan .....	8
Activity Diagram Login .....	8
Activity Diagram Pengisian FPBJ .....	9
Sequence Diagram Pemilihan Penyedia .....	9
<b>BAB 4.....</b>	<b>10</b>
<b>PENUTUP.....</b>	<b>10</b>
4.1 Perbandingan Metode AUP dan Waterfall .....	10
4.2 Perbandingan Metode AUP dan Prototype .....	11
4.3 Perbandingan Metode AUP dan RAD.....	12

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

AUP menjelaskan pendekatan yang sederhana, mudah dipahami untuk mengembangkan perangkat lunak aplikasi bisnis menggunakan teknik dan konsep agile. Agile Unified Process (AUP) mengadopsi filosofi “serial in the large” dan “iterative in the small” <sup>[1]</sup> untuk membangun sistem berbasis komputer. Dengan mengadopsi tahapan kegiatan klasik Unified Process permulaan (inception), elaborasi (elaboration), konstruksi (construction), dan transisi (transition) AUP menyediakan lapisan serial (mis., Urutan linier kegiatan rekayasa perangkat lunak) yang memungkinkan tim memvisualisasikan alur proses keseluruhan untuk proyek perangkat lunak. Namun, dalam setiap kegiatan, tim melakukan iterasi untuk mencapai agility dan untuk memberikan software increment yang bermakna kepada pengguna akhir secepat mungkin.

### 1.2 Rumusan Masalah

1. Apa itu metode Agile Unified Process (AUP)
2. Apa tujuan Agile Unified Process (AUP)
3. Apa manfaat Agile Unified Process (AUP)
4. Apa saja kelebihan dan kekurangan Agile Unified Process (AUP)
5. Apa saja perbedaan metode AUP dengan metode Waterfall, Prototype, RAD

### 1.3 Tujuan

1. Mengetahui apa itu metode Agile Unified Process (AUP)
2. Mengetahui tujuan metode Agile Unified Process (AUP)
3. Mengetahui manfaat metode Agile Unified Process (AUP)
4. Mengetahui kelebihan dan kekurangan metode Agile Unified Process (AUP)
5. Mengetahui perbedaan metode Agile Unified Process (AUP) dengan metode lainnya

# BAB 2

## PEMBAHASAN

### 2.1 Pengertian

AUP ialah framework atau base practice konseptual dalam rekayasa perangkat lunak yang menekankan pada pengulangan (iterations) dalam siklus hidup pengembangan perangkat lunak. Dengan AUP, developer dapat lebih efisien dalam menghasilkan produk perangkat lunak yang berkualitas, memenuhi kebutuhan pengguna secara tepat waktu dan tepat anggaran (Lisana, 2014).

### 2.2 Tujuan

Pada dasarnya Agile Unified Process(AUP) memiliki empat fase yang setiap bertujuan untuk :

- 2.2.1. Tahap awal (Inception)** tujuannya adalah untuk mengidentifikasi ruang lingkup awal proyek, arsitektur potensial untuk sistem, dan untuk mendapatkan pendanaan awal proyek dan penerimaan pemangku kepentingan.
- 2.2.2. Elaborasi (Elaboration)** tujuannya adalah untuk membuktikan arsitektur sistem.
- 2.2.3. Konstruksi (Construction)** tujuannya adalah untuk membangun perangkat lunak yang berfungsi secara teratur, incremental basis yang memenuhi kebutuhan prioritas tertinggi dari para pemangku kepentingan proyek.
- 2.2.4. Transisi (Transition)** tujuannya adalah untuk memvalidasi dan menyebarkan sistem ke lingkungan produksi.

### 2.3 Manfaat

Manfaat dari penggunaan metode Agile Unified Process (AUP) adalah dapat memberikan feedback kepada tim developer untuk menambahkan atau mengubah fitur dari aplikasi tersebut sebelum benar – benar dirilis

Kemudian, manajer dapat mengontrol kerja dari setiap tim dengan lebih baik. Dari segi vendor dapat mengurangi pemborosan dan dapat difokuskan pada peningkatan efisiensi

dan pengembangan fitur. Dan manfaat terakhir dari sisi developer sendiri dapat meningkatkan produktivitas tiap departemen. Karena, setiap tim dapat melakukan pengerjaan tiap tugas tanpa perlu harus menunggu tim yang lain menyelesaikan tugas nya.

## **2.4 Kelebihan dan Kekurangan**

### **1. Kelebihan AUP**

Kelebihan dari metode AUP yaitu Proses pengembangan perangkat lunak membutuhkan waktu yang relatif cepat dan tidak membutuhkan resources yang besar. Perubahan dapat ditangani dengan cepat sesuai dengan kebutuhan client. Client dapat memberikan feedback kepada tim pengembang dalam proses pembuatan program.

### **2. Kekurangan AUP**

kekurangan dari metode AUP adalah tidak cocok apabila dikerjakan oleh tim yang tidak memiliki komitmen untuk menyelesaikan proyek bersama – sama. Metode Agile ini kurang tepat apabila dikerjakan dengan jumlah skala tim yang besar (>20 orang). Tim pengembang harus selalu bersiap karena perubahan dapat terjadi sewaktu – waktu.

# BAB 3

## METODE PENGEMBANGAN SISTEM

**Kasus :** Perancangan Sistem Informasi Pengadaan Barang dan Jasa Berbasis Web

Terdapat tujuh proses yang dilakukan dan diperhatikan untuk mengawali perancangan, design model, pengembangan sistem dan validasi sistem pengadaan yang dikerjakan pada phase discipline pada gambar AUP yaitu;

1. **Model**, tujuan utama dari disiplin ini untuk memahami bisnis perusahaan, mengidentifikasi domain bermasalah yang dikerjakan dan mendapatkan solusi untuk mengatasi masalah.
2. **Implementasi**, tujuan disiplin ini untuk mengetahui source code yang tepat untuk diterapkan pada model dan dapat dijalankan kemudian dilanjutkan pengujian dasar khususnya pada pengujian modul unit.
3. **Test/uji coba**, tujuan disiplin uji coba adalah untuk melakukan evaluasi memastikan kualitas sistem, evaluasi disiplin test mencakup perhatian terhadap ada tidaknya kecacatan sistem dan melakukan test validasi sistem sudahkah bekerja sesuai perancangan dan memverifikasi persyaratan telah dipenuhi.
4. **Penyebaran Sistem**, tujuan disiplin ini adalah bagian dari rencana pengaplikasian sistem yang telah siap untuk digunakan kepada pengguna akhir.
5. **Manajemen Konfigurasi**, adalah melakukan pengelolaan yang mengatur hak akses ke dalam setiap artefak sistem pengadaan dan dapat dilakukannya control/pengendalian sistem.
6. **Manajemen Risiko**, adalah disiplin melakukan pengawasan berguna dapat mengelola sebuah resiko dengan membuat catatan atas kekurangan dan kemajuan sistem dengan menuangkan dalam keterangan laporan untuk mengidentifikasi dan mengantisipasi timbulnya masalah.
7. **Ruang Lingkup**, tujuan disiplin ini adalah untuk mendukung upaya dengan memastikan bahwa rangkaian proses, kebutuhan peralatan perangkat keras dan lunak sudah mendukung dan telah sesuai dengan standard pemangku kepentingan perusahaan dan pedoman perusahaan.

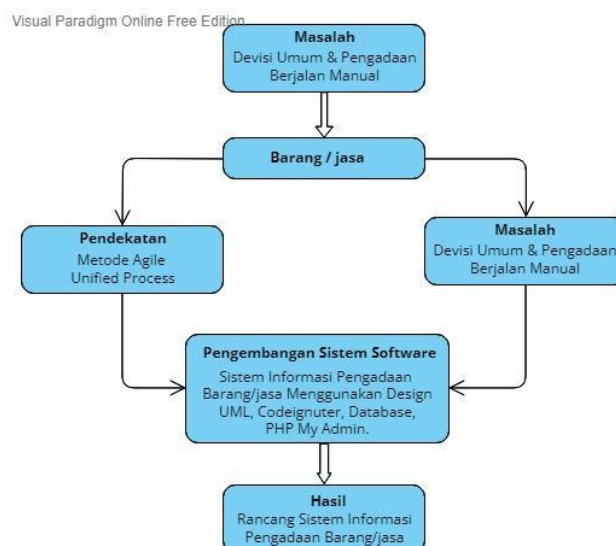
# ANALISA DAN PEMBAHASAN

## A. Perancangan

Penelitian diawali dengan menentukan kebutuhan perangkat keras untuk dapat memastikan sistem dapat berjalan dan selanjutnya menganalisa lingkup masalah dan dokumentasi data, informasi, aktivitas proses pengadaan berjalan, hal tersebut perlu dilakukan untuk mengetahui kebutuhan-kebutuhan yang diperlukan di unit kerja pengadaan. Dalam hal ini peneliti menerjemahkan keterangan tabel kebutuhan perangkat dan objek bagan alir yang berjalan.

Tabel Kebutuhan Perangkat

<i>Software</i>	<i>Hardware</i>
<ul style="list-style-type: none"><li>• <i>Operating System</i> (OS) Linux atau Windows</li></ul>	<ul style="list-style-type: none"><li>• Perangkat <i>Server</i></li><li>• Domain (jika sistem informasi akan di hosting keluar atau menggunakan <i>server</i> luar)</li><li>• Kapasitas : <i>Unlimited</i></li></ul>

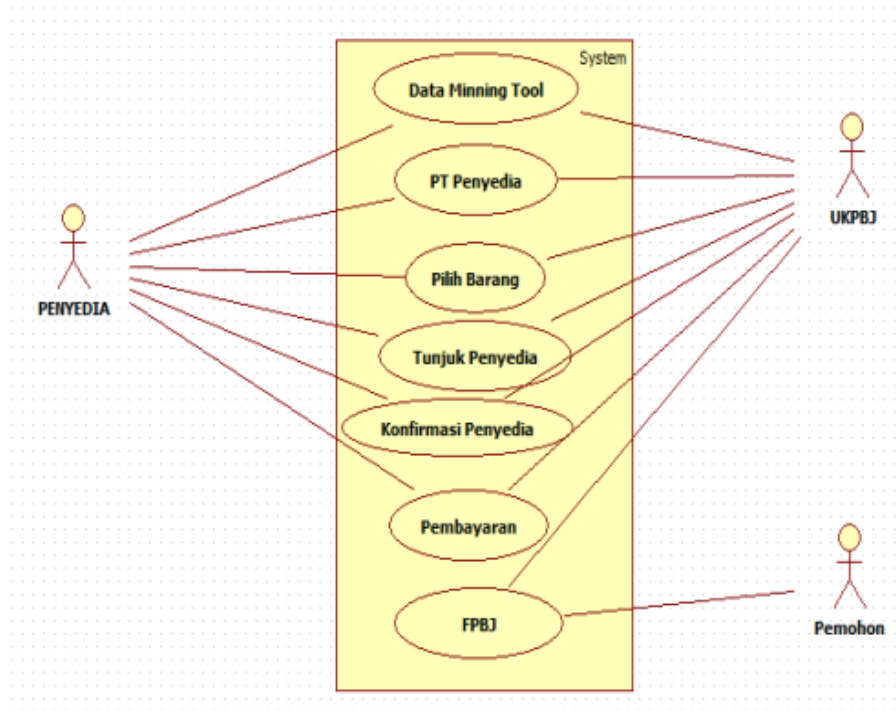


## B. Perancangan Sistem Usulan (Sistem Informasi e-Procurement)

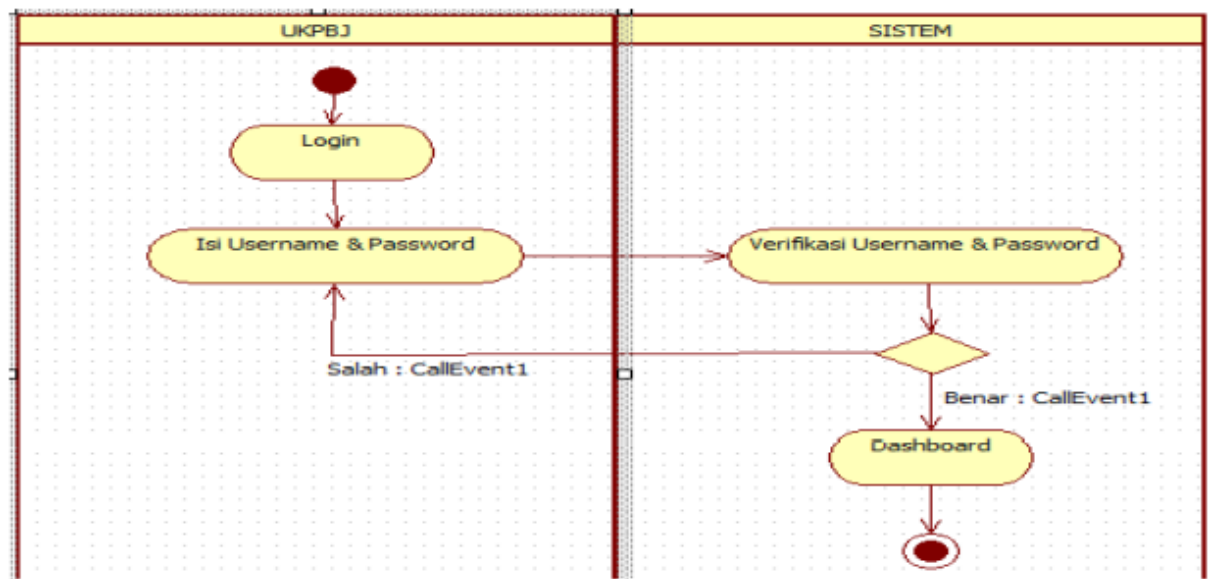
Ilustrasi dari sistem pengadaan barang/jasa yang dilakukan melalui perancangan sistem informasi eprocurement yang diusulkan oleh peneliti pada PT SMI, digambarkan

melalui Lifecycle System dan dalam diagram Model Unified Modelling Language (UML) yang terdiri dari use case, activity dan sequence.

### UC Diagram Pengadaan

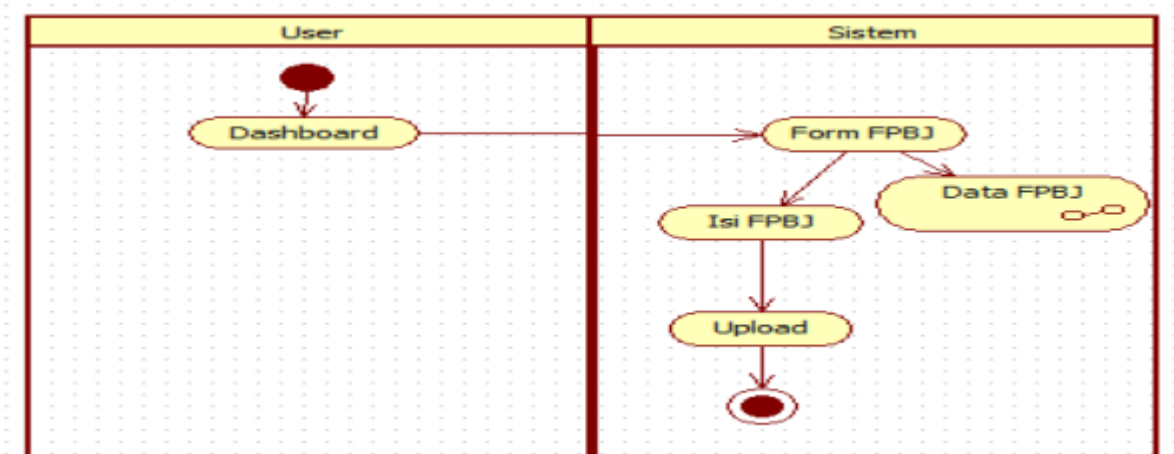


### Activity Diagram Login

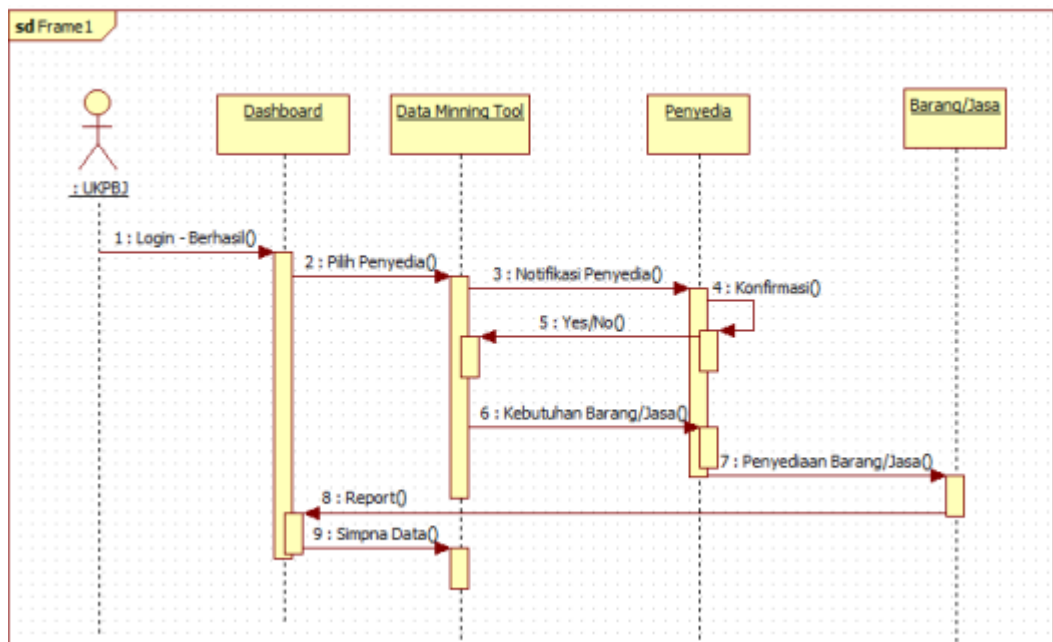




## Activity Diagram Pengisian FPBJ



## Sequence Diagram Pemilihan Penyedia



# **BAB 4**

## **PENUTUP**

### **4.1 Perbandingan Metode AUP dan Waterfall**

Perbandingan dengan metode waterfall, Prototype, RAD dan AUP. Untuk perbedaan metode AUP dan Waterfall adalah waterfall struktur dan seringkali itu bisa sangat kaku, sementara metodologi AUP dikenal karena fleksibilitasnya. Proses pengembangan perangkat lunak dibagi menjadi beberapa fase berbeda dalam model Waterfall, sementara metodologi AUP memisahkan siklus hidup pengembangan proyek menjadi sprint.

Pada model Waterfall, pengembangan perangkat lunak diselesaikan sebagai satu proyek yang kemudian dibagi menjadi beberapa fase yang berbeda, dengan setiap fase hanya terjadi satu kali selama proyek. Metodologi Agile, di sisi lain, dapat dilihat sebagai kumpulan dari banyak proyek kecil yang berbeda. Proyek yang tidak lain adalah iterasi fase yang berbeda yang bertujuan untuk meningkatkan kualitas perangkat lunak secara keseluruhan dengan umpan balik dari pengguna atau tim QA.

Semua fase pengembangan proyek seperti desain, pengembangan, pengujian, dan lain-lain selesai satu kali dalam model Waterfall, sambil menerapkan pendekatan pengembangan berulang sebagai bagian dari metodologi AUP. Perencanaan, pengembangan, pembuatan prototipe, dan fase pengembangan perangkat lunak lainnya dapat terjadi lebih dari satu kali selama proyek AUP.

Jika Anda ingin menggunakan model Waterfall untuk pengembangan perangkat lunak, Anda harus mengetahui dengan jelas semua persyaratan sebelumnya. Tidak ada ruang untuk mengubah persyaratan begitu pengembangan proyek dimulai. Metodologi Agile cukup fleksibel dan memungkinkan untuk membuat perubahan pada persyaratan, bahkan setelah perencanaan awal selesai.

Sementara metodologi Waterfall adalah proses internal dan tidak memerlukan partisipasi pengguna, pendekatan pengembangan perangkat lunak AUP berfokus pada kepuasan pengguna dan dengan demikian pada partisipasi pengguna selama fase pengembangan. Salah satu perbedaan paling penting antara metodologi pengembangan AUP dan Waterfall adalah pendekatan mereka sendiri terhadap kualitas dan pengujian. Dalam model Waterfall, fase "Pengujian" muncul setelah fase

“Bangunan”, tetapi dalam metode AUP, pengujian biasanya dilakukan bersamaan dengan pemrograman atau setidaknya selama iterasi yang sama dengan pemrograman.

Model Waterfall dapat dianggap sebagai proses sekuensial yang ketat, tetapi metodologi AUP adalah proses pengembangan perangkat lunak yang sangat kolaboratif, yang mengarah pada input tim yang lebih baik dan penyelesaian masalah yang lebih cepat.

Model Waterfall memerlukan pola pikir proyek dan berfokus secara eksklusif pada penyelesaian pengembangan proyek. AUP telah memperkenalkan pola pikir produk yang bertujuan untuk memastikan bahwa produk yang dikembangkan memenuhi kebutuhan pengguna dan dapat disesuaikan jika kebutuhan pengguna berubah.

Model Waterfall paling cocok untuk proyek dengan persyaratan yang jelas dan di mana tidak ada perubahan yang diharapkan sama sekali. Pengembangan linier mendukung proses di mana persyaratan diharapkan untuk berubah dan berkembang. Jadi, jika Anda berencana untuk mengembangkan perangkat lunak yang perlu ditinjau secara teratur dan yang harus mengikuti perkembangan teknologi dan persyaratan pengguna, AUP adalah pendekatan terbaik untuk diikuti.

## **4.2 Perbandingan Metode AUP dan Prototype**

Model prototyping merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai. Prototipe tersebut akan dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Prototype didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah prototype disebut prototyping. Bentuk dari prototype merupakan versi awal dari tahapan sebuah sistem software yang digunakan dalam mempresentasikan gambaran dari ide, eksperimen dari sebuah rancangan, mencari sebanyak mungkin masalah yang ada serta penyelesaian terhadap masalah tersebut. Sistem dengan model prototype mengizinkan pengguna agar mengetahui seperti apa tahapan sistem dibuat sehingga sistem mampu beroperasi dengan baik.

### 4.3 Perbandingan Metode AUP dan RAD

Rapid prototyping Development adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Rapid Application Development (RAD) adalah metodologi pengembangan perangkat lunak yang berfokus pada membangun aplikasi dalam waktu yang sangat singkat. Istilah ini menjadi kata kunci pemasaran yang umum menjelaskan aplikasi yang dapat dirancang dan dikembangkan dalam waktu 60--90 hari.

	Waterfall	Prototype	RAD	AUP
Kelebihan	<ul style="list-style-type: none"> <li>• Mudah dimengerti</li> <li>• Mudah digunakan</li> <li>• Requirement dari sistem bersifat stabil</li> <li>• Baik dalam manajemen kontrol</li> <li>• Bekerja dengan baik ketika kualitas lebih diutamakan dibanding dengan biaya dan jadwal /deadline</li> </ul>	<ul style="list-style-type: none"> <li>• Memberikan klien experience yang lebih awal untuk software yang akan digunakan/diperbaiki serta melengkapinya dengan feedback yang diberikan klien</li> <li>• Resiko gagal lebih rendah</li> <li>• Komunikasi antara klien dan tim pengembangan yang intens</li> <li>• Kecilnya resiko pembengkakan biaya</li> </ul>	<ul style="list-style-type: none"> <li>• Mudah mengakomodasi perubahan sistem</li> <li>• Progress development bisa diukur</li> <li>• Mengurangi waktu developmen</li> <li>• Mempercepat untuk mendapatkan feedback user</li> <li>• Cocok untuk sistem yang berbasis komponen dan terukur</li> <li>• Mudah dalam menentukan dasar sistem</li> </ul>	<p>Proses iteratif dan Incremental. Requirement dapat berubah sewaktu waktu</p> <p>Pelacakan requirement dengan melihat Backlog produk</p> <p>Keterlibatan user secara aktif</p> <p>Rilis yang lebih cepat dan berkala, fungsi dirilis setiap akhir iterasi</p> <p>Testing dilakukan setiap saat</p>
Kekurangan	<ul style="list-style-type: none"> <li>• Semua kebutuhan sistem harus diketahui dari awal</li> <li>• Integrasi sekaligus di akhir sistem</li> <li>• Testing hanya dilakukan pada setiap akhir fase, bahkan tidak jarang testing</li> </ul>	<ul style="list-style-type: none"> <li>• Mahal</li> <li>• Sangat rawan dengan adanya perubahan yang tiba tiba, karena akan mempengaruhi proyek secara keseluruhan</li> <li>• Melakukan</li> </ul>	<ul style="list-style-type: none"> <li>• Tidak cocok untuk project dengan dana yang rendah</li> <li>• Cukup beresiko</li> <li>• Butuh kerjasama</li> </ul>	<ul style="list-style-type: none"> <li>• Interaktif dengan client yang kadang terlalu berlebihan.</li> <li>• Agile sulit diimplementasikan dalam proyek yang berskala besar.</li> <li>• Waktu perencanaan proyek yang singkat,.</li> <li>• Membutuhkan</li> </ul>

	<p>hanya dilakukan di akhir proyek</p> <ul style="list-style-type: none"> <li>• Membutuhkan waktu yang cukup lama meski proyek kadang tidak terlalu besar</li> <li>• Perubahan requirement dapat merubah keseluruhan proses yang telah dilaksanakan</li> </ul>	<p>modifikasi dapat mengganggu workflow dari tim development</p>	<p>yang tinggi dan proses yang cepat</p>	<p>manajemen tim terlatih</p>
--	--	--	--	-------------------------------