



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 *Purchasing module*

Dengan adanya modul *purchasing* [10], membantu segala integrasi yang berkaitan dengan proses pembelian pada suatu perusahaan dalam mengadakan barang maupun stok dalam inventori yang dimiliki. Dalam aktivitas pembelian atau *purchasing* yang berproses mulai dari melakukan order pembelian barang, pengiriman order kepada bagian vendor penjual, dan sampai pada barang yang disorder telah dinyatakan sampai pada gudang perusahaan tersebut.

2.2 *Financial Accounting and Controlling module*

Kegiatan akuntansi dasar seperti mencakup hal-hal seperti pencatatan piutang usaha dan utang usaha, pencatatan transaksi penjualan maupun pembelian yang dilakukan perusahaan serta pencatatan aset perusahaan dapat dilakukan dalam modul ERP yaitu *Financial Accounting* (FI) [11]. Mengukur dan mengenali *economic costs* merupakan *key role* dari *financial accounting* yang dapat mengarahkan pada perkembangan dari manajemen keuangan dan memberikan keputusan yang tepat [12]. *Financial Accounting* berkaitan dengan kegiatan pelaporan informasi kepada perusahaan untuk membantu membuat keputusan ekonomi yang baik tentang kinerja perusahaan dan posisi keuangan [13]. *Controlling* hadir untuk memberikan kontribusi untuk dapat memenuhi tujuan utama perusahaan dengan meningkatkan efektivitas dan efisiensi tindakan kewirausahaan [14].

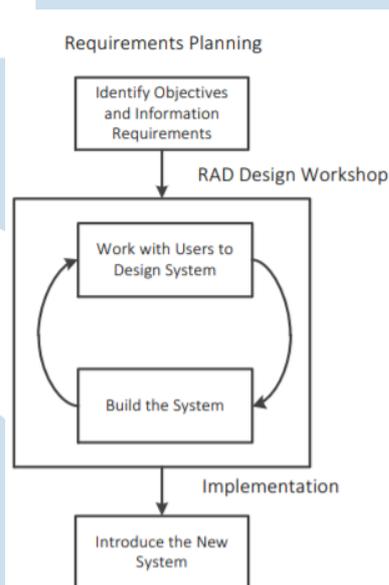
Pada perusahaan produksi, *Financial Accounting* dan *Controlling* merupakan hal terpenting dalam perusahaan tersebut dalam mengendalikan setiap *cost* yang berada dalam lingkup perusahaan [15].

2.3 Inventory Management module

Pengolahan inventori sangat diperlukan, guna untuk dapat memenuhi beberapa kebutuhan penting seperti barang-barang yang digunakan untuk melengkapi aktivitas yang dilakukan didalam perusahaan. Penerapan *inventory management* juga memberikan efisiensi dan efektifitas dalam pengelolaan inventori seperti kesiapan dalam setiap barang yang ada dalam perusahaan dengan tepat [16].

2.4 Rapid Application Development Methodology

Rapid Application Development menjadi salah satu solusi dalam proses pembuatan sistem informasi menjadi lebih cepat [17].



Gambar 2. 1 Tahapan Metode *Rapid Application Development*

Sumber: [18].

Dengan adanya 3 tahapan tersebut, pada gambar 2.1 akan menjelaskan mengenai tahapan dari metode *Rapid Application Development* [19]:

1. *Requirement Planning*

Tahapan *requirement planning* akan menjadi tahapan awal dari metode RAD, dalam tahapan ini peneliti akan mengidentifikasi tujuan

yang akan menjadi pembuatan sistem serta kebutuhan informasi mengenai bisnis yang terkait dengan permasalahan yang akan diselesaikan sistem. Keterlibatan dari *end user* dan pembuat sistem akan menjadi poin penting untuk *requirement planning*.

2. *RAD Design Workshop*

RAD Design Workshop merupakan tahapan perencanaan dan pemecahan masalah. Hasil akhir dari proses *RAD Design Workshop* adalah sistem yang akan meliputi organisasi sistem, dan struktur data. Keaktifan dan keterlibatan *end user* akan menjadi penentu dari pencapaian tujuan dari proses ini, diperlukan perbaikan yang terjadi apabila tidak sesuai dengan keinginan *end user*.

3. *Implementation*

Tahapan implementasi adalah tahapan terakhir pada metode *Rapid Application Development* (RAD). Tahapan implementasi merupakan tahapan dimana *programmer* yang mengembangkan sistem akan menyelesaikan sistem. Pengujian akan dilakukan terlebih dahulu sebelum dapat digunakan oleh *end user* nantinya. Tanggapan *end user* akan menjadi patokan bagi *programmer* untuk dapat masuk ke tahapan implementasi.

Dengan menggunakan metode *Rapid Application Development* (RAD), ada beberapa keuntungan yang akan didapatkan, kelebihan dari menggunakan *Rapid Application Development* (RAD) adalah [20]:

1. Menghemat waktu pengerjaan *project*,
2. Mengurangi kebutuhan yang berkaitan dengan biaya dan SDM dalam *project*,
3. Sudut pandang akan diambil dari sisi *user* sehingga fungsi akhir dari sistem mampu dipakai dengan mudah,
4. Menciptakan rasa kepemilikan yang kuat dalam pemegang kebijakan *project*, dan

5. Perubahan desain dapat dilakukan dengan cepat dibandingkan dengan pendekatan SDLC tradisional.

Selain adanya kelebihan yang didapatkan ada juga kekurangan dari metode *Rapid Application Development* (RAD), ada beberapa kekurangan dari penggunaan *Rapid Application Development* (RAD) adalah [20]:

1. Dengan adanya waktu yang sangat singkat, maka *project* harus diselesaikan dengan waktu yang sudah diberikan,
2. Aplikasi yang dibuat selesai dengan cepat belum menentukan bahwa aplikasi yang dibuat mampu menyelesaikan permasalahan yang ada, dan
3. Menyulitkan *programmer* yang tidak berpengalaman menggunakan perangkat dalam pengembangan sistem karena perlu keahlian dalam sistem yang digunakan.

2.5 PHP

Hypertext Preprocessor (PHP) merupakan salah satu bahasa yang digunakan untuk membuat pemrograman berbasis *web*, PHP dapat digunakan untuk *HTTP request* [21]. Direktori penyimpanan yang berisi file-file PHP dapat digunakan sebagai kelas model pada aplikasi yang dapat menyempurnakan fungsi utama dibentuknya sistem [22].

2.6 Unified Modelling Language

Unified Modelling Language (UML) merupakan salah satu metode untuk membuat permodelan visual dalam perancangan serta pembuatan sebuah *software*. UML mendefinisikan lima belas teknik diagram yang digunakan untuk memodelkan suatu sistem [23].

Terdapat dua macam pengelompokan utama dalam permodelan diagram, yaitu *behavioral diagrams* dan *structure diagrams*. *Structure diagrams* menyediakan cara untuk mewakili data dan hubungan statis dalam sistem informasi.

Structure diagrams terdiri dari *class*, *object*, *package*, *deployment*, *component*, *composite structure*, dan *profile diagrams*. Sedangkan *behavioral diagrams* akan menyediakan analisis dengan menghubungkan relasi dinamis antara *instances* atau *objects* yang akan mewakili bisnis dalam sistem. Sedangkan *behavioral diagrams* terdiri dari *activity*, *sequence*, *communication*, *interaction overview*, *timing*, *behavioral state machine*, *protocol state machine*, dan *use-case*. Pada gambar 2.2 merupakan ringkasan dari UML.

Diagram Name	Used to...	Primary Phase
Structure Diagrams		
Class	Illustrate the relationships between classes modeled in the system	Analysis, Design
Object	Illustrate the relationships between objects modeled in the system; used when actual instances of the classes will better communicate the model	Analysis, Design
Package	Group other UML elements together to form higher-level constructs	Analysis, Design, Implementation
Deployment	Show the physical architecture of the system; can also be used to show software components being deployed onto the physical architecture	Physical Design, Implementation
Component	Illustrate the physical relationships among the software components	Physical Design, Implementation
Composite Structure Design	Illustrate the internal structure of a class, i.e., the relationships among the parts of a class	Analysis, Design
Profile	Used to develop extensions to the UML itself	None
Behavioral Diagrams		
Activity	Illustrate business workflows independent of classes, the flow of activities in a use case, or detailed design of a method	Analysis, Design
Sequence	Model the behavior of objects within a use case; focuses on the time-based ordering of an activity	Analysis, Design
Communication	Model the behavior of objects within a use case; focus on the communication among a set of collaborating objects of an activity	Analysis, Design
Interaction Overview	Illustrate an overview of the flow of control of a process	Analysis, Design
Timing	Illustrate the interaction among a set of objects and the state changes they go through along a time axis	Analysis, Design
Behavioral State Machine	Examine the behavior of one class	Analysis, Design
Protocol State Machine	Illustrate the dependencies among the different interfaces of a class	Analysis, Design
Use-Case	Capture business requirements for the system and illustrate the interaction between the system and its environment	Analysis

Gambar 2. 2 Ringkasan Dari UML Diagram

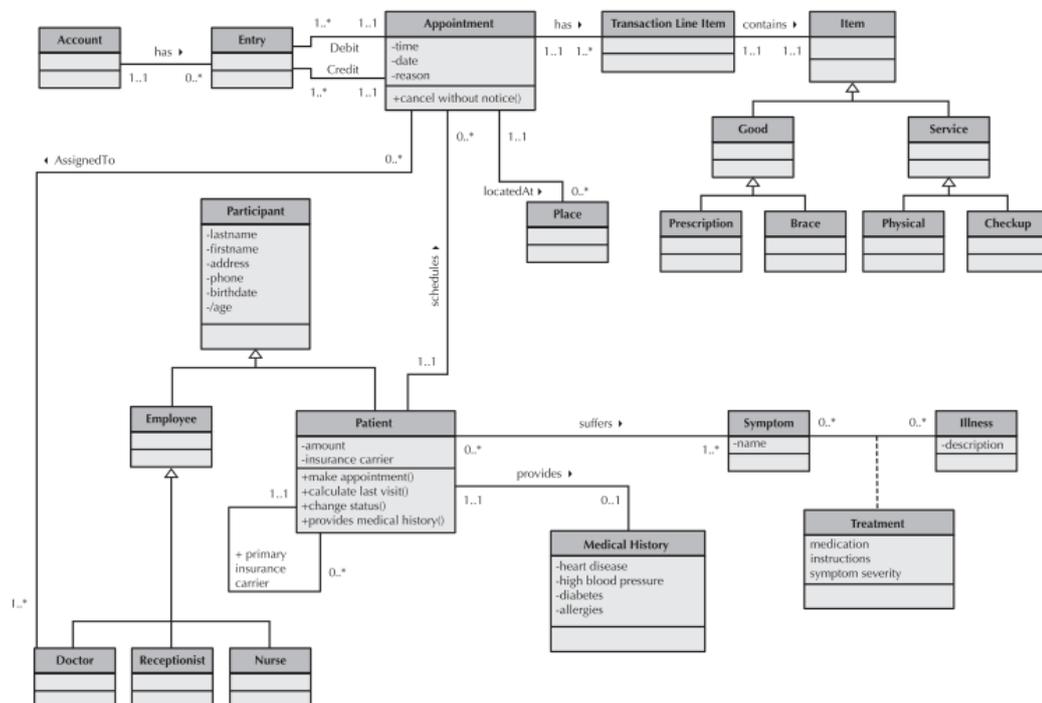
Sumber: [23].

1. Structure Diagram

Dalam struktur diagram disini menyebutkan dan menggambarkan elemen dari spesifikasi, struktur diagram tersebut terdiri atas:

1) Class Diagram

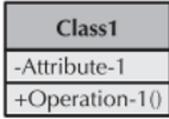
Class diagram merupakan bagaimana hubungan antar *class* yang ada dalam sistem. *Class diagram* akan menjelaskan setiap struktur dari sistem pemrograman yang dibangun. Contoh *class diagram* terdapat pada gambar 2.3 dibawah ini.



Gambar 2. 3 Contoh Class Diagram

Sumber: [23].

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Penjelasan	Notasi
<p><i>A class :</i> Mewakili jenis orang, tempat, atau hal-hal yang diperlukan oleh sistem dalam menangkap dan menyimpan informasi. Memiliki nama yang akan diketik dengan huruf tebal dan berada ditengah kompartemen atas.</p>	
<p><i>An association :</i> Merupakan hubungan antara class atau class dengan dirinya sendiri. Diberi label menggunakan kata kerja atau nama peran yang mewakili hubungan tersebut.</p>	
<p><i>A generalization :</i> Mewakili semacam hubungan beberapa kelas.</p>	
<p><i>An aggregation :</i> Mewakili hubungan bagian-bagian <i>logical</i> antar beberapa <i>class</i> atau <i>class</i> dengan dirinya sendiri, merupakan bentuk khusus dari <i>association</i>.</p>	
<p><i>A composition :</i> Akan mewakili hubungan bagian fisik antar <i>class</i> maupun dengan dirinya sendiri, dan merupakan bentuk khusus dari <i>association</i>.</p>	

Tabel 2. 1 Notasi Pada Class Diagram

Sumber: [23].

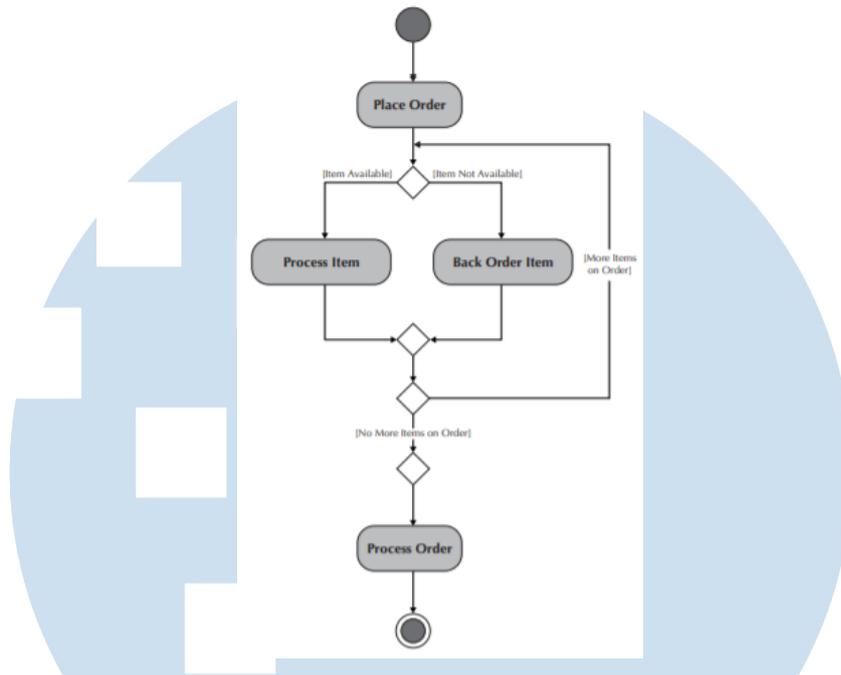
Pada tabel 2.1 merupakan penjelasan notasi-notasi yang terdapat pada *class diagram*. Penjelasan notasi ini akan memberikan informasi untuk pembuatan *class diagram* pada penelitian ini.

2. Behavior Diagram

Behavior Diagram sendiri menjelaskan fungsi sistem yang sesuai dengan *business process*. *Behavior Diagram* terdiri dari:

1) Activity Diagram

Menjelaskan setiap aktivitas objek yang ada dalam sistem, merupakan bentuk visual dari aktivitas maupun tindakan yang ada dalam sistem. *Activity Diagram* akan menggambarkan setiap perilaku sistem. Contoh *Activity Diagram* pada gambar 2.4.



Gambar 2. 4 Contoh Activity Diagram

Sumber: [23].

Penjelasan	Notasi
<p><i>An activity:</i> Merepresentasikan sebuah <i>actions</i> yang dilabeli dengan namanya.</p>	
<p><i>An object:</i> Digunakan untuk menunjukkan <i>object</i> terhubung dengan <i>flow</i> set pada <i>object</i>. Akan dilabeli dengan <i>class name</i>.</p>	
<p><i>A control flow:</i> Menunjukkan urutan dari eksekusi yang dilakukan.</p>	
<p><i>An initial node:</i> Merupakan permulaan dari aktivitas.</p>	
<p><i>A final-activity node:</i> Menandakan perhentian dari <i>object flows</i> dalam aktivitas.</p>	
<p><i>A decision node:</i> Digunakan untuk mewakili kondisi pengujian dalam aliran kontrol atau objek. Akan dilabeli dengan kriteria keputusan yang diambil untuk melanjutkan jalur aktivitas.</p>	
<p><i>A merge node:</i> Digunakan apabila dua aktivitas masuk kembali dalam suatu kondisi tertentu.</p>	

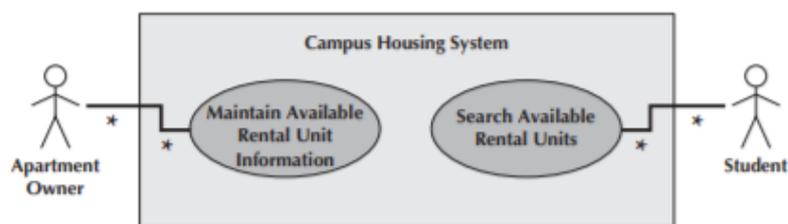
Tabel 2. 2 Notasi Pada Activity Diagram

Sumber: [23].

Pada tabel 2.2 merupakan penjelasan dari notasi yang digunakan pada *activity diagram*. Informasi mengenai notasi ini nantinya akan digabungkan untuk pembuatan *activity diagram* yang akan menjelaskan aktivitas pada setiap *object* maupun *class* yang akan berinteraksi.

2) Use Case Diagram

Use Case Diagram akan menggambarkan setiap peranan dari *actor* yang ada dalam penelitian. Contoh dari *Use Case Diagram* ada pada gambar 2.5.



Gambar 2. 5 Contoh Use-Case Diagram

Sumber: [23].

Penjelasan	Notasi
<i>An actor :</i> Merupakan orang yang melaksanakan maupun terhubung pada menangkap maupun menerima informasi dalam sistem,	 Actor
<i>An association :</i> Merupakan hubungan antara <i>use case</i> dengan <i>actor</i> .	
<i>An use case :</i> Merepresentasikan <i>major</i> pada fungsi sistem secara keseluruhan.	

Tabel 2. 3 Notasi Use-Case Diagram

Sumber: [23].

Pada tabel 2.3 merupakan notasi dari *use-case diagram*. Notasi ini akan berguna untuk perancangan pada *use-case diagram* yang akan dibuat.

2.7 HTML

HTML adalah salah satu bahasa yang digunakan untuk membentuk suatu *web*, dengan adanya kode HTML maka dapat diterjemahkan kedalam komputer pengguna dalam bentuk tampilan teks, grafik, atau multimedia [24]. Dengan adanya HTML, membantu sistem dalam melengkapi fungsi-fungsi tertentu seperti mengontrol tampilan *web* dan menambahkan objek-objek tertentu seperti suara, gambar, maupun *video* [25].

2.8 Database

Data adalah kumpulan dari beberapa kejadian yang merupakan hasil dari kejadian yang nyata, dapat menjelaskan keterangan suatu hal yang dapat berupa suatu bermakna [26]. Database adalah kumpulan dari data atau informasi yang akan disusun menjadi kelompok-kelompok tertentu berdasarkan tipe data yang sesuai [27].

2.9 MySQL

MySQL merupakan salah satu *database server* yang dapat digunakan untuk membantu pembuatan aplikasi berbasis *web* yang bersumber pada *database* untuk pengelolaannya [28]. MySQL merupakan sebuah *database management system* (DBMS) yang bersifat *open source* dengan arsitektur *client server* atau *embedded system*, dengan sifat *open source* akan sangat cocok digunakan untuk replikasi basis data [29].

2.10 XAMPP

XAMPP dapat digunakan sebagai *software web server apache* dan dapat menghubungkan dengan *server MySQL* melalui pemrograman PHP agar dapat membuat *website*, dalam XAMPP ini terdapat Apache, MySQL, dan PHP yang akan berguna untuk pembuatan *website*. XAMPP sudah dapat mendukung sistem operasi pada *windows* [30].

2.11 Laravel

Framework Laravel sendiri merupakan *software* yang dapat digunakan untuk pembuatan *web* dalam metode *web engineering*, dengan Laravel ini membantu pembuatan *web* agar lebih efektif dan efisien dengan pemrograman PHP statis [31]. Untuk membandingkan antara pembuatan *web* secara *Native* dan menggunakan *framework* Laravel [32], terdapat beberapa perbandingannya seperti pada tabel 2.4 dibawah ini.

Tabel 2. 4 Perbandingan Laravel dan PHP Native

Sumber: [32]

Perbandingan	Laravel	PHP Native
Jumlah code	Jumlah code yang dibutuhkan untuk satu fungsi cenderung pendek	Jumlah code yang dibutuhkan untuk menjalankan suatu fungsi cenderung lebih panjang
Efisiensi Kode	Dengan penggunaan laravel, menjalankan satu kode yang ingin diulang tidak perlu mengulang kodenya, hanya cukup menjalankan satu fungsi	Untuk menjalankan kode yang ingin diulang, kode tersebut harus mengalami looping terus menerus
Arsitektur	Laravel menggunakan arsitektur MVC (Model, View, and Controller) dan OOP (Object Oriented Programming)	PHP Native tidak menggunakan arsitektur MVC

2.12 Penelitian Sebelumnya

Penelitian sebelumnya dipakai untuk referensi pembuatan sistem informasi berbasis *web* yang akan dibuat. Referensi yang digunakan ada pada tabel 2.5. Referensi-referensi yang diambil pada penelitian terdahulu sebagai acuan dalam desain pembuatan sistem berbasis *web* dan pemodelan sistem serta acuan pemilihan metode dalam perancangan sistem.

Tabel 2. 5 Tabel Penelitian Terdahulu

Informasi Jurnal	Hasil Penelitian	Kesimpulan
<p>Judul Jurnal: Analisa dan Perancangan Sistem Informasi Pembelian GRC dengan Metode Waterfall [33] Penulis: H. Larasati, S. Masripah Tahun: 2017 Nama Jurnal: Jurnal Pilar Nusa Mandiri, Vol. 13, No. 2, Hal. 193-198, September 2017</p>	<p>Analisa yang dibuat menghasilkan sistem informasi pembelian untuk merubah pencatatan manual menjadi sistem dikomputerisasi.</p>	<p>Dengan penelitian ini, memberikan gambaran untuk pembuatan sistem informasi pembelian untuk Hotel yang memang masih menggunakan pencatatan pembelian secara manual.</p>
<p>Judul Jurnal: Sistem Informasi Inventori Aset Berbasis Web Pada Harris Hotel Denpasar [34] Penulis: Rikajaya, I. N., Liandana, M., & Yudiastra, P. P. Tahun: 2020 Nama Jurnal: Jurnal Teknologi Informasi dan Multimedia, Vol. 2, No. 2, Agustus 2020</p>	<p>Hasil penelitian berupa sistem informasi berbasis web untuk melihat pencatatan inventori aset pada Hotel Harris Denpasar.</p>	<p>Memberikan referensi pada pembuatan sistem informasi pencatatan inventori. Tidak hanya pencatatan inventori secara penyimpanan data melalui microsoft word saja, melainkan sistem berbasis web mampu menampilkan secara real-time untuk perubahan inventori tersebut.</p>

<p>Judul Jurnal: Studi Analisis Model Rapid Application Development dalam Pengembangan Sistem Informasi [35] Penulis: Aswati, S., Ramadhan, M. S., Firmansyah, A., & Anwar, K. Tahun: 2017 Nama Jurnal: Jurnal Manajemen: Teknik Informatika Dan Rekayasa Komputer, Vol. 16, No. 2, Page 20-27.</p>	<p>Analisis dilakukan untuk mencari alternatif metode pengembangan sistem, dengan <i>Rapid Application Development</i> dapat digunakan untuk mengurangi waktu yang digunakan untuk pengembangan sistem informasi.</p>	<p>Digunakan sebagai acuan dalam metode pengembangan sistem menggunakan metode <i>Rapid Application Development</i>.</p>
<p>Judul Jurnal: Analisis dan Perancangan Sistem Informasi Pengadaan Barang Berbasis Web di PT Anggada Husada Pertiwi [36] Penulis: Hutajulu, Brian Rheinhard Oloan Tahun: 2018 Nama Jurnal: Perpustakaan UMN, Skripsi Tahun 2018 Fakultas Teknik Informatika Program Studi Sistem Informasi</p>	<p>Hasil penelitian berupa pembuatan sistem informasi berbasis web untuk pengadaan barang.</p>	<p>Analisis yang dilakukan memberikan pandangan untuk pembuatan web yang bertujuan untuk pengadaan barang pada suatu perusahaan.</p>

<p>Judul Jurnal: Perancangan Sistem Administrasi Penyewaan Alat Kontraktor yang Berbasis Web Pada PT Megah Saputro Jaya [37] Penulis: Saputro, Dody Try Tahun: 2018 Nama Jurnal: Perpustakaan UMN, Skripsi Tahun 2018 Fakultas Teknik Informatika Program Studi Sistem Informasi</p>	<p>Hasil penelitian berupa pembuatan sistem berbasis web untuk penyewaan alat kontraktor</p>	<p>Penelitian tersebut memberikan contoh pembuatan web untuk sistem penyewaan barang dan pencatatan transaksi pada perusahaan</p>
<p>Judul Jurnal: Sistem Informasi Berbasis Web Untuk Penjualan Solar Pada PT. Sahassa Prima Niaga [38] Penulis: Saputra, Danny Halim Tahun: 2018 Nama Jurnal: Perpustakaan UMN, Skripsi Tahun 2018 Fakultas Teknik Informatika Program Studi Sistem Informasi</p>	<p>Hasil penelitian berupa sistem informasi berbasis web untuk penjualan solar</p>	<p>Sebagai acuan referensi dalam desain pembuatan sistem berbasis web.</p>