

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan peneliti ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan peneliti yang dilakukan. Beberapa uraian peneliti terdahulu yang menjadi acuan dalam penelitian ini yaitu :

1. Menurut Stephanie Surja, dkk, 2014, penelitian ini menghasilkan suatu sistem informasi produksi yang membuat perancangan aplikasi permintaan. Penelitian tersebut merancang dan mengimplementasikan Sistem Informasi untuk menangani proses-proses penjadwalan produksi, perencanaan bahan baku, pencatatan keluar masuknya barang dari gudang.
2. Nova Sagita, dkk (2015), dengan judul "Perancangan Sistem Informasi Produksi Rajut Menggunakan Metode System Development Life Cycle (SDLC) (Study Kasus : CV. Suho Germino Bandung). Penelitian ini bertujuan untuk menyelaraskan aliran sistem informasi dan merancang sistem informasi produksi rajut di CV. Suho Garmino. Sistem informasi produksi rajut membutuhkan langkah yang harus dilakukan agar aliran sistem tersebut digunakan untuk merancang sistem informasi ini adalah model System Development Life Cycle (SDLC). Sedangkan pada tahap analisis dan perancangan sistem dilakukan dengan pendekatan terstruktur.

3. Menurut Helmina, Herry Mulyono, 2017, penelitian ini menghasilkan rancangan *prototype* sistem informasi produksi yang diharapkan dapat membantu pihak PT. Putra Sumber Utama Timber (PSUT) dalam mempercepat pengelolaan datanya yang berkaitan dengan beberapa laporan yaitu laporan data admin, data *supplier*, bahan baku, proses produksi dan data hasil produksi, sedangkan penelitian yang sedang dilakukan adalah tentang bagaimana produksi pakan ikan dan pakan tambak udang yang nanti prosesnya akan menjadi sebuah makanan pelet.
4. Menurut Aji Saiful Anwar (2017), dengan judul “Sistem Informasi Produksi Plastik Pada UD. Bagas Mulya Mejobo Kudus Berbasis Web” Penelitian ini bertujuan untuk membuat atau membangun suatu aplikasi sistem informasi produksi plastik. Dengan pembuatan aplikasi ini diharapkan dapat membantu dalam setiap proses manajemen data di UD. Bagas Mulya dan mengurangi biaya pembuatan aplikasi.
5. Menurut Bustomi Romdhon (2016), dengan judul “Perancangan Sistem Informasi Produksi Pada PT. Selectrix Indonesia” Sistem informasi produksi usulan yang dirancang dapat merubah kegiatan proses produksi yang awalnya serba manual menjadi terintegrasi dengan komputer, sehingga semua masalah kesalahan data dan keterlambatan penyebaran informasi dapat diperbaiki dan memiliki tingkat kinerja yang lebih baik dari sistem perusahaan saat ini. Penilaian tersebut didapatkan dari hasil evaluasi kinerja sistem informasi produksi usulan memiliki nilai 4,60 sedangkan sistem saat ini hanya memperoleh nilai 2,66. Sistem informasi produksi usulan dapat

membuat komunikasi yang baik antar departemen yang terkait dalam proses produksi.

II.2. Landasan Teoritis

II.2.1. Sistem Informasi

Sistem Informasi adalah merupakan kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian, yang bermaksud menata jaringan komunikasi yang penting, pengolahan atas transaksi-transaksi tertentu dan rutin, membantu manajemen dan pemakai internal dan eksternal untuk menyediakan dasar pengambilan keputusan yang tepat (Bustomi Romdhon, dkk, 2016).

II.2.1.2 Sistem Informasi Produksi

Di dalam berbagai macam pembicaraan, dikatakan bahwa produksi adalah merupakan dapurnya perusahaan tersebut. Menurut Hendra Kusuma (2004 ; 4) “Produksi diartikan sebagai kegiatan yang dapat menimbulkan tambahan manfaat atau penciptaan faedah baru. Faedah atau manfaat ini dapat erdiri dari beberapa macam, misalnya faedah bentuk, faedah waktu, faedah tempat serta kombinasi dari faedah-faedah tersebut diatas. Apabila terdapat suatu kegiatan yang dapat menimbulkan manfaat baru, atau mengadakan penambahan dari manfaat yang sudah ada, maka kegiatan tersebut akan disebut sebagai kegiatan produksi, di dalam hal ini tidaklah menjadi persoalan” (Helmina, dkk, 2017).

II.2.3. Pakan Ikan dan Pakan Tambak Udang

Pakan ikan merupakan salah satu faktor yang berperan penting dalam proses pertumbuhan ikan. Pertumbuhan ikan dapat berjalan optimal apabila jumlah pakan, kualitas pakan dan kandungan nutrisi terpenuhi dengan baik dan merupakan sumber nutrisi bagi ikan yang dapat menghasilkan energi untuk aktivitas hidup. (Rohmad Zaenuri, dkk, 2016).

Pakan Udang merupakan salah satu faktor yang sangat menentukan dalam keberhasilan budidaya udang. Udang merupakan salah satu jenis binatang yang cukup potensial untuk dibudidayakan sebagai usaha. Salah satu di antaranya adalah pemberian pakan yang efektif dan efisien.

II.2.4. Metode System Development Life Cycle (SDLC)

Menurut Sukamto dan Shalahuddin (2013:26) mengemukakan bahwa “SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan *best practice* atau cara-cara yang sudah teruji baik.

Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri daritahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni: siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan prototyping (*life cycle using prototyping*) dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*).

Proses tahapan SDLC yang paling sering digunakan adalah :

1. Perencanaan

Mempelajari konsep sistem permasalahan yang hendak diselesaikan, apakah sistem baru tersebut realistis dalam masalah pembiayaan, waktu, serta perbedaan dengan sistem yang ada sekarang.

2. Analisis Sistem

Menganalisis konsep sistem, permasalahan dan keperluan yang hendak dibuat ke dalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi hambatan yang terjadi.

3. Desain

Tahap setelah analisis sistem dari siklus pengembangan sistem yang mendefinisikan dari kebutuhan fungsional, persiapan untuk rancang bangun implementasi.

4. Konstruksi

Suatu kegiatan membangun sarana maupun prasarana, perbaikan terhadap produk yang memiliki kesalahan/kerusakan.

5. Implementasi

Software yang telah diuji dan siap diimplementasikan kedalam sistem pengguna/ sudah siap diterapkan.

6. *Maintenance*

Sistem yang telah diimplemantasikan serta dapat mengikuti perkembangan dan perubahan apapun yang terjadi guna meraih tujuan penggunaannya.

SDLC atau System Development Life Cycle atau sering disebut juga System Development Life Cycle adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan best practice atau cara-cara yang sudah teruji baik” (Yoki Firmansyah, dkk, 2018).

Siklus hidup pengembangan Sistem atau SDLC adalah metodologi untuk merancang, membangun, dan memelihara informasi dan proses sistem. Terdapat banyak model SDLC, salah satunya adalah model Waterfall yang terdiri dari lima tahap untuk secara berurutan diselesaikan dalam rangka untuk mengembangkan solusi perangkat lunak (Nurasiah, 2014).

System Development Life Cycle (SDLC) adalah keseluruhan proses dalam membangun sistem melalui beberapa langkah. Ada beberapa model SDLC. Model yang cukup populer dan banyak digunakan adalah waterfall. Beberapa model lain

SDLC misalnya fountain, spiral, rapid, prototyping, incremental, build & fix, dan synchronize & stabilize (Isnardi, 2016).

II.2.5. SQL Server 2008

SQL (Structure Query Language) merupakan bahasa *query* standar yang dipergunakan untuk mengakses basis data relasional . *Statement SQL* secara garis besar dibagi menjadi 3 ketegori yaitu *Data Definition Languages (DDL)*, *Data Manipulation Languages (DML)* dan *Data Control Languages (DCL)*. Dalam *SQL Server* juga dikenal dengan istilah *RDBMS (Relational Database Management System)*. *RDBMS* adalah dasar untuk *SQL*, dan untuk semua sistem *database modern* seperti *Microsoft SQL Server*, *IBM DB2*, *Oracle*, *MySQL*, dan *Microsoft Access*. Data dalam *RDBMS* disimpan dalam objek *database* yang disebut tabel (Fitri Marisa, 2015).

II.2.6. Microsoft Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, aplikasi *Console*, aplikasi *Windows*, aplikasi *Web*. Visual Studio mencakup compiler, *SDK*, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic.NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe* (Herpendi, 2016).

II.2.7. Normalisasi

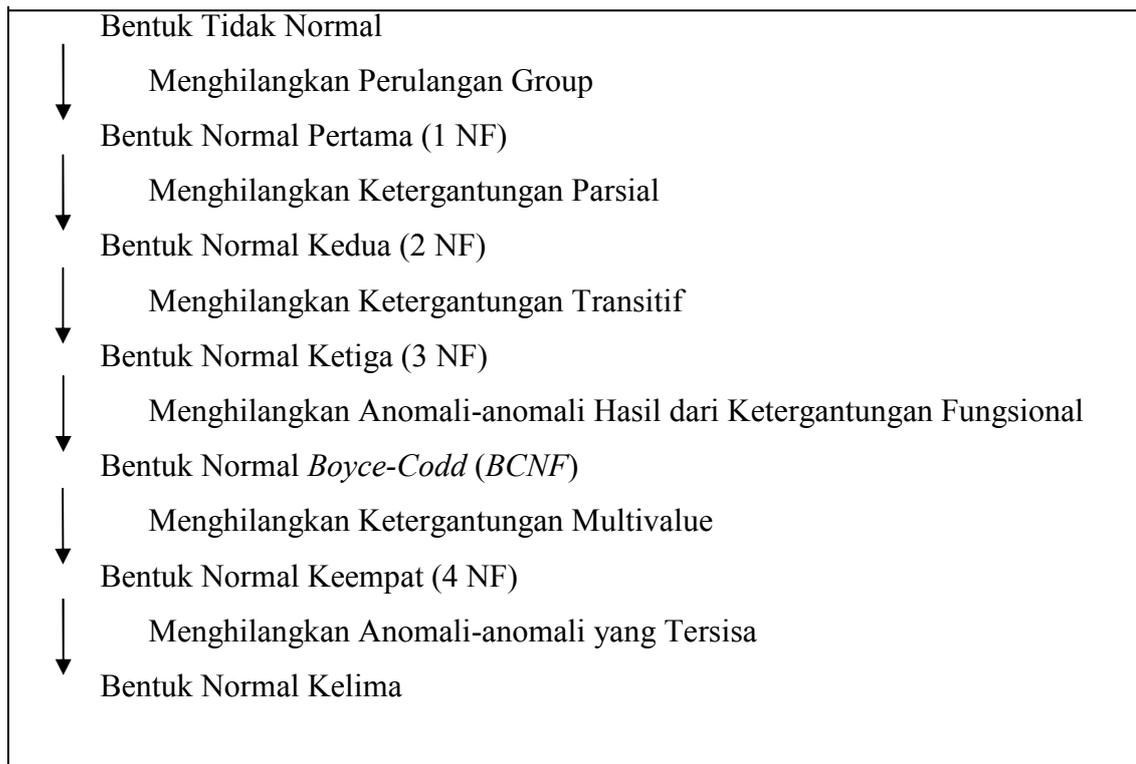
Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh base relations (*file*).
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Adapun proses normalisasi sebagai berikut :

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat kemudian.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan. Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi dapat dilihat pada Gambar II.1.



Gambar II.I. Tahapan Normalisasi
 (Sumber : Dwi Puspitasari, et al : 2016)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan/ didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd* Normal Form (BCNF) Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Pada

penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu 3NF. (Dwi Puspitasari, et al : 2016).

II.2.8. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2012 : 95).

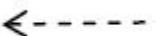
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

a. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi siapa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan

fungsi-fungsi tersebut. Simbol - simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel II.1.

Tabel II.1. Simbol - simbol Use Case Diagram

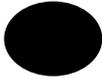
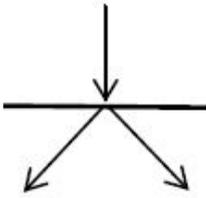
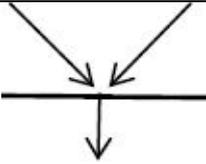
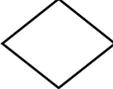
Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem, Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini, Vol 4 : 2016)

b. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Simbol - simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel II.2.

Tabel II.2. Simbol - simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

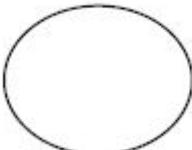
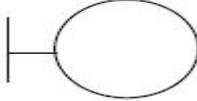
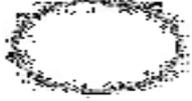
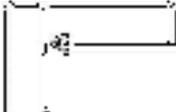
(Sumber : Ade Hendini, Vol 4 : 2016)

c. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima

objek. Simbol - simbol yang digunakan dalam *sequence diagram* dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol – simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Bounadry Class</i> , berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan form cetak.
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan objek.
	<i>Message</i> , Simbol pengiriman antar pesan.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirm untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktiivtas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, Vol 4 : 2016)

II.14.4. *Class Diagram*

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan

dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: kelas (*class*), Relasi, *Associations*, *Generalization* dan *Aggregations*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti. Simbol - simbol yang digunakan dalam *class diagram* dapat dilihat pada Tabel II.4.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau satu atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal
n..n	Batasan antara, Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, Vol 4 : 2016)