

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian dari David Yudhi Kurniawan, 2019, yang berjudul “Rancang Bangun Sistem Informasi Pengendalian Persediaan Barang dengan Metode *Economic Order Quantity* (EOQ) Berbasis Dekstop Dengan Menggunakan Java Netbeans 8.2 pada Wijaya Celluler”. Tujuan dari penelitian ini adalah mempermudah pegawai dalam pengolahan pengendalian barang dengan menggunakan sistem informasi pengendalian persediaan barang dengan metode *economic order quantity* (EOQ) pada Wijaya Celluler. Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model waterfall. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis mulai dari tahap analisis, desain, kode, pengujian, dan pemeliharaan. Hasil penelitian ini adalah sistem informasi pengendalian persediaan barang dengan metode *economic order quantity* (EOQ) pada Wijaya Celluler.

Menurut Navin Ariwidya Pratama¹, 2019, yang berjudul “Perancangan Aplikasi Perencanaan Bahan Baku Menggunakan Metode MRP (*Material Requirement Planning*) Pada PT. E-T-A Indonesia” Penelitian ini bertujuan untuk membuat suatu aplikasi yang dapat membantu memudahkan dan memaksimalkan kinerja pegawai administrasi gudang dalam meramalkan jumlah barang yang harus

dipesan ke pabrik untuk periode berikutnya. Metode peramalan yang digunakan adalah metode kuantitatif trend projection dengan variasi data musiman. Metode ini membutuhkan informasi data pada tahun-tahun sebelumnya sehingga pada penelitian ini mengambil data 3 tahun sebelumnya. Dengan metode peramalan ini diperoleh hasil peramalan nilai permintaan hampir mendekati dengan data sebenarnya. Dari hasil pengujian sistem yang dikenakan pada 5 data permintaan diperoleh tingkat akurasi sistem sebesar 86%..

Menurut Asvin Wahyuni, 2015, yang berjudul “ Perencanaan Persediaan Bahan Baku Dengan Menggunakan Metode *Material Requirement Planning* (MRP) Produk Kacang Shanghai Pada Perusahaan Gangsar Ngunut-Tulungagung” Penelitian ini menyimpulkan bahwa total biaya persediaan bahan baku tahun 2012 dengan menggunakan metode perusahaan yang digunakan sebelum penelitian adalah sebesar Rp 50.063.563.595,-. Setelah dilakukan penelitian dengan menggunakan metode *Material Requirement Planning* (MRP) mengalami penurunan sebesar Rp 1.072.427.967,- artinya perusahaan dapat meminimalisasikan biaya persediaan sebesar 46,7 %. Dari perbedaan total biaya persediaan sebelum dan sesudah penelitian menunjukkan bahwa metode MRP dapat diterapkan pada perusahaan “Gangsar” sehingga perencanaan bahan baku dapat berjalan secara efektif dan efisien.

Menurut Maria Yosefin Amelia, 2019, yang berjudul “ Perancangan Sistem Basis Data Persediaan Bahan Baku Berbasis Metode Economic Order Quantity (EOQ) (Studi Kasus: PT Malindo Intitama Raya)” Penelitian ini bertujuan untuk mengetahui jumlah persediaan bahan baku yang tepat serta mengembangkan suatu

sistem basis data yang dapat mendukung pengendalian persediaan bahan baku di PT Malindo Intitama Raya. Dengan adanya sistem basis data yang terkomputerisasi dengan baik, pengambilan keputusan dapat dilakukan dengan lebih cepat dan tepat karena didukung oleh data-data yang dapat diakses dengan cepat

Menurut Pudji Astuti, 2018, yang berjudul “ Perancangan Sistem Informasi Perencanaan dan Pengendalian Persediaan Bahan Baku di PT. X” Penelitian ini bertujuan untuk menjadwalkan pembelian bahan baku dengan mempertimbangkan *lead-time* waktu pengiriman, *lead-time* waktu produksi dan jumlah bahan baku yang tersedia di gudang. Dengan sistem MRP pembelian bahan baku akan terencana, terjadwal, tidak mendadak, tidak ada permintaan yang terselip dan bahan baku dipesan pada jumlah yang optimum. Sistem informasi mengatasi ketidakpastian *lead-time* pengiriman dan ketidakpastian permintaan dengan menerapkan stok pengaman. Stok pengaman digunakan sebagai persediaan cadangan yang akan diproduksi selama bahan baku belum datang atau saat terjadi permintaan yang mendadak Sistem informasi mampu mengendalikan ketersediaan bahan baku. Pada sistem informasi terdapat sistem peringatan dini yang mana akan memberikan warning kepada ketersediaan bahan baku sudah dibawah stok pengaman. Sistem ini berfungsi untuk menjaga bahan baku tetap pada level yang aman.

II.2. Uraian Teoritis

II.2.1. Sistem Informasi

Sistem informasi merupakan sistem yang disusun secara sistematis dan teratur dari jaringan-jaringan aliran informasi yang menghubungkan setiap bagian

dari suatu sistem, sehingga memungkinkan diadakannya komunikasi antar bagian atau satuan fungsional. (Agus Irawan : 2 : 2016)

Sistem informasi ini terdiri dari kata sistem dan informasi. Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Sedangkan kata informasi berasal dari kata Perancis kuno *informacion* (tahun 1387) yang diambil dari bahasa Latin *informationem* yang berarti “garis besar, konsep, ide”. Informasi merupakan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Zaenal Sekty Wijaya: 2 : 2015).

II.2.1.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen sebagai berikut :

1. Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

Blok model ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila

terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi. (Tomi Loveri : 2 : 2018)

2.2.1.2 Tujuan Perancangan Sistem Informasi

Pada umumnya tujuan melakukan suatu perancangan sistem informasi adalah (Eka Iswandi : 3 : 2015) :

1. Untuk memenuhi kebutuhan pemakai sistem informasi.
2. Untuk memberikan gambaran yang jelas dan rancang bangun lengkap kepada pemrogram komputer dan ahli teknik lain yang terlibat.
3. Untuk mendukung pengolahan pelaporan manajemen dan mendukung perusahaan.

2.2.1.3. Alat Bantu Dalam Perancangan Sistem Informasi

Dalam perancangan sistem informasi memiliki alat bantu perancangan sistem yaitu (Eka Iswandi : 2015) :

1. Aliran Sistem Informasi (ASI)

Merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Untuk itu dibutuhkan pedoman– pedoman untuk membuat Aliran Sistem Informasi (ASI).

2. *Context Diagram* (CD)

Context diagram merupakan *data flow diagram* yang menggambarkan garis besar operasional sistem. Konteks diagram menggambarkan hubungan sistem dengan entitas-entitas di luar sistem. CD memperlihatkan sistem sebuah proses.

Tujuannya adalah memberikan pandangan umum sistem. CD memperlihatkan sebuah proses yang berinteraksi dengan lingkungan luarnya. Ada pihak luar yang memberikan masukan dan pihak yang menerima keluaran sistem.

3. *Data Flow Diagram (DFD)*

Diagram aliran data sistem disebut juga dengan *Data Flow Diagram (DFD)*. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan. DFD menggambarkan arus data didalam sistem dengan terstruktur dan jelas.

4. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) memiliki dua komponen utama yaitu Entitas (*Entity*) dan Relasi (*Relation*). Kedua komponen ini ,masing-masing dilengkapi dengan sejumlah atribut yang mempresentasikan seluruh fakta yang ada di dunia nyata.

5. *Flowchart*

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis.

II.2.2. Metode *Material Requirement Planning*

Material Requirement Planning (MRP) adalah sebuah sistem informasi terkomputerisasi yang dikembangkan secara spesifik untuk membantu dalam

pengelolaan persediaan untuk permintaan dependen dan penjadwalan ulang pesanan. Istilah-istilah dalam MRP yang digunakan adalah:

1. *Gross Requirement* (Kebutuhan Kotor) adalah total demand yang didapatkan dari semua rencana produksi induk termasuk kebutuhan yang diantisipasi untuk setiap periode waktu.
2. *Schedule Receipts* Merupakan jumlah item yang akan diterima pada suatu periode tertentu berdasarkan pesanan yang dibuat.
3. *Project On Hand Inventory* adalah jumlah persediaan yang tersedia setelah semua kebutuhan kotor terpenuhi.
4. *Net Requirement* merupakan jumlah aktual yang diinginkan untuk diterima atau diproduksi dalam periode tertentu.
5. *Planned Order Receipt* perencanaan penerimaan untuk menjaga persediaan tidak dibawah nol.
6. *Planned Order Releases* adalah kapan pesanan dengan jumlah tertentu akan dipesan (Pudji Astuti : 2018)

Heizer dan Render (2018) menyebutkan bahwa MRP adalah model permintaan terikat yang menggunakan daftar kebutuhan bahan, status persediaan, penerimaan yang diperkirakan, dan jadwal produksi induk, yang dipakai untuk menentukan kebutuhan material yang akan digunakan.

Roger G. Schroeder (2018) menyebutkan MRP sebagai suatu sistem informasi yang digunakan untuk merencanakan dan mengendalikan persediaan dan kapasitas. MRP merupakan komputerasi sistem persediaan seluruh bahan yang dibutuhkan

dalam proses konversi suatu perusahaan, baik usaha manufaktur maupun usaha jasa.

Berdasarkan definisi yang dikemukakan oleh beberapa pakar yang dimaksud di atas, maka MRP dapat diartikan sebagai sebuah metode perencanaan dan pengendalian material (bahan baku, parts, komponen, dan subkomponen) yang terikat pada unit produksi yang akan dihasilkan, dengan menggunakan suatu sistem yang sudah terintegrasi.

II.2.3. Metode *Economic Order Quantity* (EOQ)

Economic Order Quantity (EOQ) merupakan sistem pengendalian persediaan yang menggunakan matematika dan statistik sebagai alat bantu utama dalam memecahkan masalah kuantitatif. Metode ini bertujuan untuk menentukan mengoptimalkan ukuran pemesanan ekonomis (EOQ), titik pemesanan kembali (ROP) dan jumlah cadangan pengaman (*safety stock*, SS) yang diperlukan. (Maria Yosefin Amelia : 2019).

EOQ adalah suatu metode yang digunakan untuk pengendalian barang pada gudang untuk meminimalkan biaya pemesanan dan penyimpanan barang dalam periode tertentu. Perhitungan EOQ yaitu:

$$EOQ = \sqrt{\frac{2DS}{H}} \dots\dots\dots(1)$$

Keterangan:

EOQ = Jumlah optimum unit per pesanan

S = Biaya pesanan untuk setiap pesanan

D = Permintaan tahunan dalam unit

H = Biaya penyimpanan per unit per tahun (David Yudhi Kurniawan : 2019)

II.2.4. 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, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, *Software Development Kit* (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).

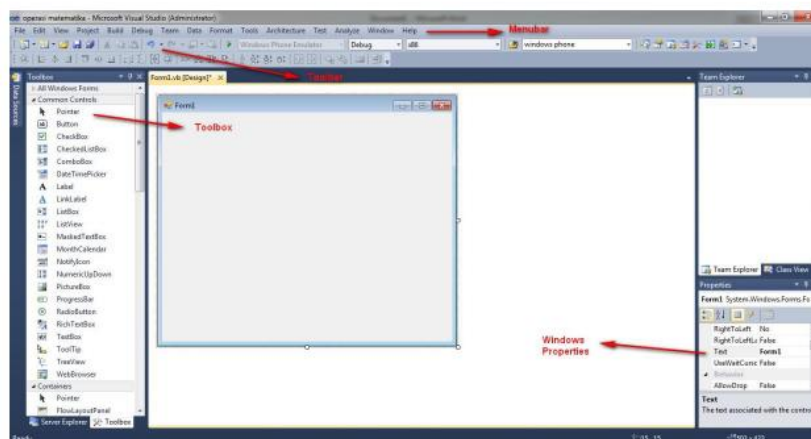
Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). (Herpendi, 2016).

Visual Studio sebelumnya versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk *platform* Microsoft *.NET Framework 3.5*. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform *.NET Framework 2.0*

dan 3.0. Visual Studio 2003. ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998). (Herpendi, 2016).

II.2.4.1 Komponen dalam VB 2010

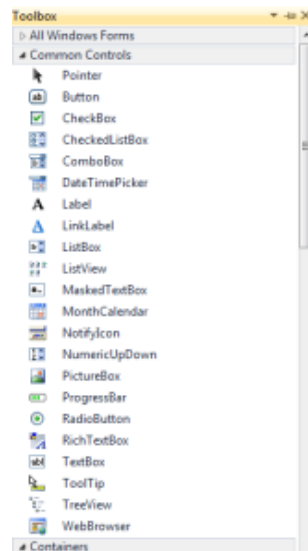
Jendela Visual Basic atau sering juga disebut lingkungan kerja Visual Basic mempunyai tampilan yang hampir sama dengan tampilan jendela program aplikasi Windows yang sudah kita kenal, seperti Word, Excel, dan Power Point. Di aplikasi visual basic terdapat menu-menu dan toolbar yang memuat icon-icon dan tombol-tombol untuk menjalankan perintah-perintah. Perbedaannya, Visual Basic mempunyai beberapa tambahan komponen, yaitu **Toolbox**, **Windows Project**, dan **Windows Properties**.



Gambar II.1. Komponen Visual Studio 2010
(Sumber : Wahyu Trisno Atmojo : 6 : 2018)

Toolbox ditempatkan disebelah kiri jendela kerja. Window Project dan Window Properties berada disebelah kanan jendela kerja. Jika kita akan membuat sebuah project

baru, maka pada bagian tengah jendela akan terdapat sebuah form. Form tersebut akan digunakan sebagai tampilan (jendela) untuk program yang dibuat.



Gambar II.2. Toolbox Visual Studio 2010
(Sumber : Wahyu Trisno Atmojo : 9 : 2018)

II.2.5. SQL Server

SQL Server merupakan suatu *Relational Database Management Systems* (RDBMS) yang digunakan untuk menyimpan data. Data yang disimpan pada *database* bisa dalam skala kecil maupun besar. Selain itu, penyajiannya merupakan penyajian pada level fisik karena kita akan menyimpan langsung data pada *database* dengan kondisi yang sebenarnya, yaitu disimpan pada tabel apa, kolom mana, dan menggunakan tipe data saat penyimpanan. (Benardo, 2015).

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query*. Pada *SQL Server 2008* terdapat fitur-fitur yang dapat mengembangkan performa dari *database* tersebut. Beberapa fitur tersebut, yaitu (Benardo, 2015) :

- 1) *Date Data Type*: Digunakan untuk menyimpan data tanggal saja sehingga akan menghemat *space* pada *server*.
- 2) *Data Compression*: Digunakan untuk melakukan compress data sehingga ukuran data yang disimpan dalam hal *space hardisk* akan lebih kecil.
- 3) *Sparse Column*: Digunakan untuk menyimpan data yang memiliki lebih banyak data NULL dengan lebih efisien.
- 4) *Row Constructor*: Digunakan untuk melakukan insert beberapa data sekaligus dengan satu perintah INSERT.
- 5) *Table-Valued Parameter*: Digunakan untuk melakukan parsing array pada bahasa pemrograman, dimana satu variable diberikan data-data yang akan diproses setelahnya.

Microsoft SQL server 2000 adalah Sistem manajemen basis data yang memakai perintah *Transact-SQL* untuk mengirim perintah dari Komputer client ke Komputer server. Microsoft SQL Server 2000 dapat dijalankan pada NT 4.0 *Server* atau *Microsoft Windows 2000 Server*, selain itu dapat pula di install pada personal desktop di *Windows 2000 Profesional*, *Windows 98 Milenium*. Sistem akan dipergunakan oleh beberapa komputer (*client-server*) maka database yang dibangun adalah merupakan database yang berfungsi untuk menunjang hal tersebut untuk itu dalam penulisan ini dipakai database *Microsoft SQL Server 2000*, selain itu database ini juga compatible dengan bahasa pemrograman yang digunakan. Secara teoritis, program *SQL* yang sudah terinstall pada komputer dapat menampung *32.767 database* dan terdapat lebih dari *2 billion objek*. *SQL Server* adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk

aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah *client* adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Sedangkan sebuah *server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

SQL Server menggunakan tipe dari *database* yang disebut database relasional. Database relasional adalah database yang digunakan sebuah data untuk mengatur atau mengorganisasikan kedalam tabel. Tabel-tabel adalah alat bantu untuk mengatur atau mengelompokan data mengenai subyek yang sama dan mengandung informasi dan kolom dan baris. Tabel-tabel saling berhubungan dengan mesin *database* ketika dibutuhkan. *SQL Server* mendukung beberapa tipe data yang berbeda, termasuk untuk karakter, angka, tanggal (*datetime*) dan uang (*money*), *SQL Server* digunakan untuk menggambarkan model dan implementasi pada *database*. (Hendra Kurniawan : 2 : 2017).

II.2.6. Database

Konsep Database Pangkalan data atau basis data (bahasa Inggris: *database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut

sistem manajemen basis data (database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi (Benardo, 2015).

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query*. Selain itu juga dalam mengambil data dari server lain akan mengalami penurunan performa. Tetap dengan menggunakan terdistribusi, bisa dengan cepat melakukan akses untuk data pada database server yang didistribusikan. Sedangkan untuk tersentralisasi, karena databasenya hanya satu dan terpusat (misalnya di *head office*) maka seluruh client dari manapun akan mengambil data tersebut dari satu database. Dengan demikian data yang diambil tidak akan bermasalah dalam hal konsistensi karena berada dalam satu sumber, tetapi akan membutuhkan hardware yang jauh lebih besar dan bandwidth yang lebih tinggi. Hal ini dikarenakan server tersebut berfungsi untuk menampung penggunaan connection yang sangat banyak. (Benardo, 2015).

Database menurut Bambang Hariyanto dalam (Minarni & Susanti, 2014) adalah :”kumpulan data (elementer) yang secara logic berkaitan dalam mempresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam system tertentu”. (Ade Putra, 2016).

Menurut Sutabri (2016) dalam buku Sistem Informasi Manajemen: ‘Database adalah suatu kumpulan data terhubung (interrelated data) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (controlled redundancy’. (Fitri Ayu : 2018)

Beberapa model *Database* diantaranya :

1. *Object based* data model merupakan himpunan data dan prosedur atau relasi yang menjelaskan hubungan logis antar data dalam suatu *database* berdasarkan objek datanya.
2. *Record Based* data model. Model ini berdasarkan pada *record* untuk menjelaskan kepada *user* tentang hubungan logis antardata dalam *database*.

II.2.6.1 Langkah-langkah Perancangan *Database*

Langkah-langkah yang dilakukan untuk perancangan basis data :

1. Menentukan kebutuhan *file* basis data untuk sistem baru, hal ini ditunjukkan oleh *data store* pada diagram aliran data (DAD) sistem.
2. Menentukan parameter *file* basis data. Parameter *file* basis data meliputi tipe *file*, nama atribut, tipe dan ukuran, serta kunci relasi.
3. Normalisasi *file* basisdata. Langkah ini dimaksudkan untuk pengujian pada setiap file

II.2.7 UML (*Unified Modelling Language*)




Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.




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

II.2.7.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel II.1. Simbol Use Case Diagram

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukaran pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>Use Case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mendefinisikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem biasa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>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.</p>

	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat dipenuhi.

(Sumber : Ade Hendini, 2016)

II.2.7.2. Class Diagram

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Tabel II.2. Simbol Class Diagram




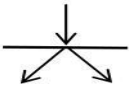
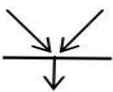


Multiplicity	Penjelasan
1	Satu dan hanya satu
0...*	Boleh tidak ada atau 1 atau lebih
1...*	1 atau lebih
0..1	Boleh tidak ada maksimal 1
n..n	Batasan antara Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2016)

II.2.7.3. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

Tabel II.3. Activity Diagram




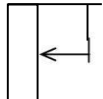


Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (Percabangan) digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel 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, true, false
	<i>Swimlane</i> pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa

(Sumber : Ade Hendini, 2016)

II.2.7.4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>Entity Class</i>, merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system dan menjadi landasan untuk menyusun basis data</p>
	<p><i>Boundary Class</i> berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem. Seperti tampilan <i>formentry</i> dan <i>form</i> cetak.</p>
	<p><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 berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar kelas.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>

(Sumber : Ade Hendini, 2016)