BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Arif Rizky Pamungkas dengan judul ‘Sistem Informasi *Inventory* Bahan Bakar Alternatif (Residu) di PT. Duta Surya Mulia Berbasis Web’ bertujuan untuk merancang sistem informasi *inventory* berbasis web pada sistem pergudangan yang ada di PT. Duta Surya Mulia. Kebutuhan akan bahan bakar alternatif (Residu) menjadikan PT. Duta Surya Mulia memberikan pilihan alternatif bahan bakar untuk mesin-mesin yang ada di pabrik. Namun dikarenakan dengan permasalahan yang ada di sistem pergudangan PT. Duta Surya Mulia, seperti pencatatan dan penghitungan data barang yang masuk dan keluar masih dilakukan secara manual, yakni lewat buku, menjadi potensi masalah yang menyebabkan sistem pergudangan yang kurang efektif dan efisien. Pemenuhan kebutuhan untuk bagian produksi menjadi tuntutan sistem pergudangan untuk dapat terkelola dan terorganisir dengan baik sehingga sistem informasi inventory dapat diharapkan menjadi bagian dari pemenuhan keputusan dalam sistem pergudangan untuk melakukan pengecekan dan pembelian barang agar selalu mencukupi stok yang ada di gudang [1].

Persamaan peneliti ini dengan peneliti yang dilakukan oleh Arif Rumahorbo adalah, peneliti sama-sama melakukan analisis pada sistem

pergudangan serta objek penelitian yang sama-sama memiliki masalah pada pencatatan yang masih dilakukan secara manual.

Perbedaan permasalahan yang terjadi dalam PT. Duta Surya Mulia adalah bagian pergudangan yang belum melakukan pembuatan laporan, yang menyulitkan bagian pergudangan untuk mengambil keputusan pada pengecekan dan pembelian barang untuk memenuhi stok yang ada di pergudangan. Sedangkan di dalam Delisa Delicious, mereka sudah mempunyai pembuatan laporan, namun masih dilakukan secara manual sehingga bagian pergudangan masih kesulitan dalam mengecek dan membeli barang untuk memenuhi stok yang ada di pergudangan.

Kemudian, penelitian yang dilakukan oleh Elang Putra Pratama dengan judul ‘Sistem Informasi Pergudangan Pada PT. Bhanda Graha Reksa Cabang Manado’. Pentingnya pengelolaan pergudangan dan juga pendataan yang masuk dan keluar dalam gudang menjadi vital bagi perusahaan yang bergerak di bidang logistik. Dikarenakan persaingan bisnis logistik juga yang memicu rancangan dari sistem informasi pergudangan yang ada di PT. Bhanda Graha Reksa Cabang Manado [2].

Persamaan peneliti ini dengan peneliti yang dilakukan oleh Elang Putra Pratama adalah, peneliti sama-sama melakukan analisis pada sistem pergudangan.

Perbedaan permasalahan yang terjadi dalam PT. Bhanda Graha Reksa Cabang Manado adalah pencatatan barang yang ada di gudang

dilakukan dengan menggunakan *software* Microsoft Excel. Sedangkan didalam Delisa Delicious, pencatatan masih dilakukan secara manual.

2.2. Landasan Teori

2.2.1. Sistem Informasi

Sistem informasi, menurut Robert A. Leitch dan K. Roscoe Davis, merupakan sistem yang berjalan di dalam organisasi untuk mengakselerasi proses bisnis yang ada di dalam organisasi. Sistem informasi menggabungkan kebutuhan untuk pengolahan transaksi harian, membantu kegiatan operasional, dan memberikan laporan kepada pihak luar tertentu yang berkepentingan. Selain itu, sistem informasi juga berfungsi sebagai kegiatan manajerial dan merupakan aktivitas strategis organisasi. Menurut John Burch dan Gary Grudnitski, komponen dalam sistem informasi dikelompokkan berdasarkan blok bangunan (*building block*) sebagai berikut. Dalam Delisa Delicious, sistem informasi akan diimplementasikan sebagai media yang dapat membantu melancarkan kegiatan operasional.

1. **Blok Masukan**

   Blok ini mewakili data yang masuk ke dalam sistem informasi, dimana data dimasukkan atau ditangkap dengan media atau metode-metode tertentu. Dalam Delisa Delicious, data yang akan masuk ke dalam sistem informasi, dapat berupa dokumen dan catatan harian dari kegiatan operasional yang dilakukan.

2. **Blok Model**

Blok ini mewakili pemodelan yang akan mengolah atau memanipulasi data yang masuk dari blok masukkan, dengan menggunakan kombinasi prosedur, logika, dan model matematika. Blok model ini akan memanipulasi data dari catatan dan dokumen yang masuk untuk diolah menjadi sebuah informasi.

3. **Blok Keluaran**

Blok ini akan mengeluarkan produk atau informasi dari data yang sudah diolah atau dimanipulasi dalam blok model, dan menjadikan informasi tersebut berkualitas dan dapat digunakan oleh semua tingkatan manajemen dan pemakai sistem. Ketika data dokumen dan catatan sebelumnya sudah diolah, akan menghasilkan informasi yang dapat digunakan untuk melakukan pengambilan keputusan, seperti pemenuhan stok yang ada di gudang dalam Delisa Delicious.

4. **Blok Teknologi**

Blok ini merupakan bagian yang mengoperasikan 3 blok utama sebelumnya, yakni blok masukan, blok model, dan blok keluaran. Blok ini dapat menerima masukkan data, menjalan model yang sudah dirancang, serta mengeluarkan informasi yang sudah diolah atau dimanipulasi oleh model yang digunakan. Blok ini memiliki 3 bagian utama seperti perangkat keras (hardware), perangkat lunak (software), dan teknisi atau operator (*humanware* atau *brainware*). Dalam Delisa Delicious, *hardware* seperti mesin komputer, *keyboard* dan *mouse* dapat digunakan untuk memasukkan

dan menyimpan data ke dalam sebuah *software* atau sistem yang dijalankan dalam mesin komputer tersebut, yang dioperasikan oleh karyawan Delisa Delicious.

5. **Blok Basis Data**

Blok ini merupakan bagian yang menyimpan dan/atau menyerahkan data yang tersimpan dalam basis data dari/ke sistem informasi. Basis data merupakan kumpulan data yang saling berhubungan, yang disimpan dalam perangkat keras, dan dimanipulasi dengan menggunakan perangkat lunak. Basis data ini dapat digunakan untuk menyimpan dokumen dan catatan untuk dimasukkan ke dalam sistem yang dirancang, sehingga dokumen dan catatan dapat tersimpan sebagai arsip ataupun diolah menjadi informasi untuk pengambilan keputusan.

6. **Blok Kendali**

Blok ini merupakan bagian yang mengendalikan sistem informasi, agar sistem informasi dapat berjalan sesuai dengan prosedur dan metode yang telah ditentukan oleh teknisi atau operator [3]. Misalkan, adanya rancangan untuk mengendalikan penggunaan bahan baku yang ada di gudang dalam Delisa Delicious.

2.2.2. Metode Pengembangan Air Terjun (*Waterfall*)

Metode pengembangan air terjun atau *waterfall* merupakan salah satu model pengembangan siklus hidup pengembangan sistem (SDLC). Metode pengembangan *Waterfall* ini digunakan untuk menjadi pemandu

perancangan sistem monitoring dalam Delisa Delicious. Tahapan-tahapan dalam metode pengembangan air terjun adalah sebagai berikut.

1. **Analisis Kebutuhan Perangkat Lunak**

   Tahapan ini merupakan proses mengetahui kebutuhan *user* untuk dirancang dan diimplementasikan dalam perangkat lunak yang akan dibuat secara intensif. Setiap kebutuhan *user* didokumentasikan untuk menghasilkan spesifikasi kebutuhan perangkat lunak. Dalam tahapan ini, penulis melakukan pemgambilan data dengan wawancara dengan karyawan yang ada di Delisa Delicious untuk merancang sistem yang sesuai dengan yang dibutuhkan dalam Delisa Delicious.

2. **Desain**

   Tahapan ini menerjemahkan kebutuhan *user* yang ada di dalam spesifikasi kebutuhan perangkat lunak menjadi sebuah desain yang menggambarkan kebutuhan *user* tersebut, seperti desain struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean perangkat lunak. Setelah penulis melakukan pengambilan data dengan wawancara, informasi kebutuhan kemudian digunakan untuk mendesain dan menyusun kerangka program agar sesuai dengan kebutuhan dalam Delisa Delicious.

3. **Pembuatan Kode Program**

   Tahapan ini kemudian menerjemahkan apa yang sudah dirancang dari tahapan desain untuk langsung mengkodekan perangkat lunak, yang

hasilnya berupa program komputer yang sesuai dengan desain dan spesifikasi kebutuhan *user*. Dalam tahapan ini, desain dan kerangka program yang sudah disusun untuk kemudian diterjemahkan menjadi kode program yang akan membentuk suatu sistem aplikasi.

4. **Pengujian**

Tahapan ini melakukan pengujian dari perangkat lunak yang sudah dikodekan sebelumnya dalam tahapan pembuatan kode program. Tahapan ini memastikan bahwa tidak adanya error dari segi logika dan segi fungsional agar keluaran program sesuai dengan yang diinginkan. Setelah sistem aplikasi sudah dibentuk dari kode-kode program sebelumnya, maka dilakukan pengujian untuk memastikan sistem aplikasi sesuai dengan spesifikasi dan kebutuhan yang diketahui dari tahap analisis.

5. **Pemeliharaan (*Maintenance*)**

Tahapan ini merupakan tahap akhir dimana perangkat lunak dikirimkan ke *user* untuk digunakan oleh *user*. Tidak menutup kemungkinan program mengalami masalah ketika sudah digunakan oleh *user*. Sehingga perlu adanya tim pemeliharaan program yang akan menganalisis kembali masalah-masalah perangkat lunak tersebut sampai pada siklus pengujian, namun siklus ini tidak sampai memunculkan perangkat lunak baru [4]. Dalam tahapan ini, aplikasi sistem yang sudah lulus dalam pengujian kemudian digunakan dan diserahkan kepada tim perawatan untuk memastikan aplikasi sistem berjalan dalam proses bisnis yang ada.

2.2.3. Metode Pendekatan Berorientasi Objek

Metode adalah prosedur atau tata cara dalam melakukan sesuatu. Sedangkan metodologi adalah ilmu yang mempelajari bagaimana prosedur atau tata cara dalam melakukan sesuatu [5]. Sedangkan pendekatan berorientasi objek adalah cara bagaimana memandang suatu rekayasa perangkat lunak atau sistem yang menggunakan model-model yang dikelola dan mengombinasikan struktur dara dan perilaku pada suatu entitas. Kumpulan objek-objek yang saling bekerja sama ini diatur oleh perilaku (*behaviour*). Dalam penelitian ini, penulis menggunakan pendekatan berorientasi objek sebagai panduan bagi penulis untuk melakukan analisis dan rancangan sistem monitoring bahan baku yang ada di Delisa Delicious. Konsep dasar dalam berorientasi objek adalah sebagai berikut.

1. **Kelas (*Class*)**

   Kelas adalah kumpulan objek yang memiliki karakteristik yang sama. Kelas terdiri dari definisi statis dan himpunan objek yang sama yang dapat muncul atau dibuat dalam kelas tersebut. Misalkan kelas pergudangan merupakan kelas yang berisi informasi mengenai pergudangan.

2. **Objek (*Object*)**

   Objek adalah suatu abstrak yang mewakili dunia nyata, seperti benda, manusia, organisasi, tempat, kejadian, struktur, status, atau hal abstrak lainnya. Objek juga memiliki kemampuan untuk menyimpan informasi (status) dan melakukan operasi (kelakuan), yang keduanya dapat

diterapkan atau mempengaruhi status dari objek tersebut. Dari kelas pergudangan sebelumnya, objek ini dapat berisi karyawan dari bagian *Purchasing* untuk mengelola dari pergudangan ini sendiri.

3. Metode (*Method*)

Metode atau operasi yang ada dalam metodologi berorientasi objek hampir sama dengan fungsi atau prosedur yang ada di metodologi struktural. Kelas yang ada di metodologi berorientasi objek dapat memiliki lebih dari satu metode atau operasi. Metode disini bisa berasal dari kejadian, aktivitas, fungsi atau kelakuan dunia nyata. Misalnya dari kelas pergudangan tadi, metode ini berisi fungsi yang dapat dilakukan oleh pergudangan itu sendiri, seperti menerima pemenuhan bahan baku dari *supplier*.

4. Atribut (*Attribute*)

Atribut adalah variabel global yang dimiliki oleh sebuah kelas. Atribut dapat berupa nilai atau elemen data yang dimiliki oleh objek dalam kelas objek itu sendiri. Dari kelas pergudangan tadi, atribut ini dapat berisi apa saja yang dapat diterima dan disimpan di gudang.

5. Abstraksi (*Abstraction*)

Abstraksi merupakan prinsip yang merepresentasikan dunia nyata yang kompleks menjadi sebuah bentuk model yang sederhana sehingga mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan yang ada. Misalkan, ketika bahan baku yang ada di pergudangan menipis seperti ketika bahan bakar kendaraan sudah mau habis.

6. Enkapsulasi (*Encapsulation*)

Enkapsulasi merupakan proses pembungkusan atribut data dan metode atau operasi yang dimiliki objek agar tidak dapat diketahui cara kerjanya oleh objek lain dengan cara menyembunyikan implementasi dan objeknya. Misalkan ketika bagian produksi hanya mengetahui stok yang ada digudang dan tidak mengetahui transaksi pemenuhan bahan baku oleh bagian *Purchasing* kepada *supplier*.

7. Pewarisan (*Inheritance*)

Pewarisan merupakan proses dimana sebuah objek dapat mewarisi sebagian atau seluruh definisi dari objek lain sebagai bagian dari diri objek itu sendiri. Misalkan ketika aktivitas dalam bahan baku tidak hanya dikelola oleh bagian *Purchasing* dalam gudang, tapi diakses dan digunakan oleh bagian produksi juga.

8. Antarmuka (Interface)

Adanya kemiripan antara antarmuka dan kelas. Namun yang membedakannya adalah antarmuka tidak mempunyai atribut kelas dan memiliki metode yang dideklarasikan tanpa isi, sedangkan kelas mempunyai atribut. Antarmuka biasanya digunakan agar kelas yang lain tidak langsung mengakses ke suatu kelas, melainkan mengakses antarmukanya terlebih dahulu. Misalnya ketika sistem aplikasi baru diakses oleh operator, yang diakses pertama kali itu adalah dashboard yang dapat menjembatani operator untuk melakukan operasi yang ada di bagian pergudangan.

9. Penggunaan Kembali (*Reusability*)

Adanya penggunaan kembali objek yang sudah dideklarasikan sebelumnya untuk masalah yang satu ke masalah lainnya yang melibatkan objek yang sama. Hal ini mencegah pembuatan objek yang sama untuk permasalahan yang satu yang bentrok dengan penggunaan objek yang sama pada permasalahan lain. Misalkan objek bahan baku tidak hanya dipakai oleh bagian *Purchasing* untuk dikelola dalam gudang, tapi digunakan oleh bagian produksi untuk bahan bakunya dipakai oleh mereka.

10. Generalisasi dan Spesialisasi

Generalisasi dan spesialisasi merupakan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Seperti pengelompokan bahan baku yang sifatnya padat dan cair. Sifat yang padat dapat berupa tepung terigu, dan sifat yang cair seperti saus cokelat.

11. Komunikasi Antar Objek

Antar objek yang ingin berkomunikasi satu sama lain dilakukan dengan mengirim pesan (*message*) yang dikirim dari satu objek ke objek yang lain. Seperti objek bagian *Purchasing* mengirimkan pesan ke objek *Supplier* untuk melakukan pemenuhan stok bahan baku yang ada di gudang.

12. Polimorfisme (*Polymorphism*)

Polimorfisme adalah ketika sebuah objek memiliki kemampuan untuk digunakan dengan nama yang sama pada banyak tujuan. Seperti bagian *Purchasing* menggunakan objek bahan baku untuk dikelola di

gudang, dan bagian produksi menggunakan objek tersebut untuk dipakai dalam produksi di Delisa Delicious.

13. Kontainer (*Package*)

Kontainer atau disebut sebagai *package* merupakan sebuah wadah yang dapat digunakan untuk mengelompokkan kelas-kelas yang ada, dan dapat digunakan untuk menyimpan kelas yang mempunyai nama yang sama dalam *package* yang berbeda-beda [4]. Contohnya seperti pengambilan dan penerimaan bahan baku di gudang dikelompokkan menjadi kontainer "Gudang" yang dapat dikelola oleh objek bagian *Purchasing*.

2.2.4. *Unified Modelling Language* (UML)

Unified modelling language merupakan bahasa pemodelan perangkat lunak yang digunakan pada pengembangan perangkat lunak dengan metode berorientasi objek. UML ini digagaskan untuk memenuhi kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. Dalam penelitian ini, penulis menggunakan UML sebagai panduan dalam analisis dan merancang sistem *monitoring* bahan baku di Delisa Delicious. Diagram dalam UML dijelaskan sebagai berikut.

1. *Use Case Diagram*

Diagram *use case* diperuntukkan untuk memodelkan perilaku (*behaviour*) dari sistem informasi yang akan dibuat. Ada dua hal utama yang ada dalam diagram *use case*, seperti Aktor dan *Use Case* itu sendiri. Aktor

merupakan seseorang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Sedangkan *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit yang akan bertukar pesan antar unit lain atau dengan aktor. Dalam penelitian ini, penulis menggunakan *Business Use Case* untuk menganalisis sistem yang berjalan, dan *System Use Case* untuk merancang sistem *monitoring* bahan baku yang ada di Delisa Delicious.

2. Skenario *Use Case*

Skenario *use case* merupakan alur jalannya proses dari unit *Use Case* yang ada didalam diagram. Skenario *use case* dibuat per unit *use case* yang menjelaskan aksi aktor dan reaksi sistem dari aksi aktor tersebut. Skenario *Use Case* di penelitian ini digunakan untuk menggambarkan aksi dalam *Business Use Case* untuk analisis dan *System Use Case* untuk merancang.

3. *Activity Diagram*

Activity Diagram atau Diagram Aktivitas merupakan diagram yang menggambarkan alur kerja dari suatu sistem atau proses bisnis yang ada. *Activity Diagram* ini menggambarkan aktivitas sistem, bukan dari apa yang dilakukan oleh aktor. *Activity Diagram* disini untuk menggambarkan apa yang sudah ada dalam analisis ataupun rancangan dalam skenario *use case* sebelumnya.

4. *Class Diagram*

Diagram Kelas atau *Class Diagram* menggambarkan struktur sistem yang mendefinisikan kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Pendefinisian kelas-kelas ini akan memudahkan pembuat program atau *programmer* untuk menstrukturkan program agar sesuai dengan desain dan kebutuhan *user* yang sudah dilakukan sebelumnya. Terdapar beberapa jenis kelas, seperti kelas *main* yang menjadi fungsi awal saat sistem dijalankan, kelas *view* yang mendefinisikan dan mengatur tampilan sistem ke pemakai, kelas *controller* yang diambil dari definisi unit fungsi yang ada di *Use Case*, serta kelas *model* yang mendefinisikan data untuk dibungkus atau digunakan dari basis data. *Class Diagram* digunakan untuk mendefinisikan kelas-kelas yang akan diterjemahkan kedalam kode program, untuk merancang sistem *monitoring* bahan baku di Delisa Delicious.

5. *Sequence Diagram*

Diagram Alur atau *Sequence Diagram* merupakan diagram yang menggambarkan kelakuan objek pada *use case* dari segi waktu hidup suatu objek dan pesan (*message*) yang dikirimkan dan diterima antar objek. Diagram alur ini akan digunakan untuk menghitung waktu response dan waktu hidup dari aktivitas atau pesan antar objek dalam rancangan sistem *monitoring* bahan baku di Delisa Delicious.

6. *Deployment Diagram*

Deployment Diagram menggambarkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram ini digunakan untuk memodelkan sistem tambahan seperti menggambarkan perangkat-perangkat yang digunakan, sistem *client*/*server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi [4]. *Deployment Diagram* digunakan untuk menggambarkan implementasi sistem *monitoring* bahan baku di Delisa Delicious.

2.2.5. *Web*

Web atau *World Wide Web* adalah sajian dan layanan informasi yang berjaringan di seluruh dunia, dimana pengguna internet mengakses dengan mudah. Layanan informasi yang disajikan umumnya memiliki tautan (*link*) pada yang berkelanjutan selama pengguna menjelajah internet, yang isinya berasal dari suatu komputer lain yang terhubung ke jaringan internet [6]. Dalam penelitian ini, penulis menggunakan *Web* sebagai wadah untuk antarmuka sistem yang akan dioperasikan oleh operator, seperti bagian *Purchasing* untuk mengelola bahan baku di gudang.

2.2.6. Basis Data

Basis data dapat didefinisikan sebagai sebuah kumpulan data yang terorganisir dan antar data mempunyai keterkaitan secara logis. Data dalam basis data dapat bervariasi dari berbagai ukuran dan kompleksitas datanya [7]. Dalam basis data, terdapat proses dan pemodelan yang dilakukan, seperti pembuatan diagram hubungan entitas, pemodelan basis data

relasional, dan proses normalisasi. Dalam penelitian ini, basis data digunakan untuk menyimpan data dari catatan dan dokumen dan menyimpan informasi yang sudah diolah dari data sebelumnya untuk diakses oleh pemakai sistem. Dijelaskan sebagai berikut.

1. **Diagram Hubungan Entitas (*Entity Relationship Diagram*)**

   Diagram Hubungan Entitas atau *Entity Relationship Diagram* merupakan pemodelan awal yang digunakan untuk memetakan entitas yang terlibat pada basis data dan saling berhubungan. Hasil dari diagram hubungan entitas ini akan digunakan untuk melakukan pemodelan basis data relasional [4]. Diagram hubungan entitas dapat digunakan untuk menggambarkan bagaimana entitas data berhubungan untuk dapat dibaca oleh sistem dan menyimpan data dari sistem yang dirancang.

2. **Model Relasional (*Relational Model*)**

   Model Relasional atau *Relational Model* merupakan model yang digunakan untuk mengkomunikasikan rancangan basis data dari perancang kepada pengguna akhir atau *end user*, dan yang paling sering digunakan selama melakukan proses analisis pengembangan basis data. Model ini merepresentasikan struktur dan kendala dari basis data yang dirancang dan arsitektur data yang akan digunakan untuk mengimplementasikan basis data [7]. Model relasional ini akan membentuk sebuah tabel relasi. Tabel relasi itu sendiri merupakan sekelompok tabel yang saling berkaitan dan berhubungan satu sama lain yang akan menggambarkan hubungan data antar tabel [8]. Model relasional dapat berupa desain yang sudah dirancang

dalam diagram hubungan entitas yang kemudian dapat digunakan untuk komunikasi dengan pemakai sistem seperti bagian *Purchasing*.

3. Normalisasi

Normalisasi merupakan kumpulan elemen-elemen data yang akan membentuk suatu tabel dan tabel-tabel ini akan menggambarkan relasi dan entitas yang terkait dari tabel yang terhubung satu sama lain [9]. Normalisasi juga dapat diartikan sebagai suatu proses tahapan yang menggunakan teknik pemodelan untuk menghasilkan struktur basis data yang baik dari segi pemanfaatan data. Normalisasi disini digunakan untuk merancang basis data yang efisien agar dapat meminimalisir terjadinya penggandaan data dalam rancangan sistem *monitoring* bahan baku di Delisa Delicious. Proses dan tahapan normalisasi dijelaskan sebagai berikut.

1. Bentuk Normal Pertama

Dalam bentuk normal pertama, proses normalisasi memastikan bahwa tidak adanya atribut jamak atau atribut ganda, dan hanya terdapat nilai tunggal yang ada di setiap pertemuan baris dan kolom dalam tabel.

2. Bentuk Normal Kedua

Dalam bentuk normal kedua, setiap dependensi fungsional parsial di hilangkan, menjadikan atribut yang bukan kunci di identifikasi oleh keseluruhan kunci primer.

3. Bentuk Normal Ketiga

Dalam bentuk normal ketiga, setiap dependensi transitif telah dihapus, yang menjadikan atribut yang bukan kunci di identifikasi oleh hanya satu kunci primer.

4. Bentuk Normal *Boyce-Codd*

Dalam bentuk normal *Boyce-Codd* ini, adanya anomali yang tersisa dihapus meskipun dependensi fungsionalnya juga dihapus. Hal ini menghindari adanya lebih dari satu kunci primer untuk atribut bukan kunci.

5. Bentuk Normal Keempat

Dalam bentuk normal keempat ini memastikan basis data terbebas dari dependensi jamak atau ganda.

6. Bentuk Normal Kelima

Dalam bentuk normal kelima ini memastikan basis data terbebas dari semua jenis anomali yang ada [7].