

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Tata kelola TI menggunakan *framework* COBIT 4.1 telah banyak diteliti dan hasil penelitiannya telah banyak membantu instansi memperbaiki tata kelola teknologi informasi dengan baik.

Rujukan penelitian tata kelola teknologi informasi terkait yang pernah dilakukan untuk mendukung penulisan antara lain:

1. Irania Dwi Wijayanti (2016) dengan judul *Analisis Maturity Level* Proses Teknologi Informasi Menggunakan *Framework* COBIT 4.1 di BSI UMY. Hasil penelitian ini menunjukkan tingkat kematangan (*maturity level*) yang ada pada setiap proses TI yang terdapat dalam *domain Plan and Organise 3* (PO3). Penelitian tersebut dilakukan dengan pengisian kuisioner oleh responden di BSI UMY dan *maturity value* dihitung dengan menggunakan Microsoft Excel.
2. Goldi Mahardika Muhammad (2017) dengan judul *Pengukuran Maturity Level* Proses TI AI3, AI4, dan DS4 Menggunakan COBIT 4.1 Pada Dinas Kominfo Bantul. Penelitian menghasilkan kesimpulan bahwa Dinas Kominfo Bantul telah mengoperasikan infrastruktur TI dalam setiap kegiatan, namun belum membuat laporan-laporan kegiatan. Analisis pengukuran *maturity level* tersebut dihitung menggunakan *software* Microsoft Excel.
3. Jani Widiastuti (2017) dengan judul *Pengukuran Maturity Level* Proses TI di BSI UMY Menggunakan *Framework* COBIT 4.1. Kesimpulan yang dihasilkan adalah BSI UMY belum mempunyai indikator yang jelas terkait keberhasilan dari penerapan perencanaan strategi TI sebelumnya. Pengukuran tersebut dilakukan menggunakan *software* Microsoft Excel.

4. Noor Azizah (2017) yang melakukan penelitian Audit Sistem Informasi menggunakan *framework* COBIT 4.1 Pada E-Learning UNISU Jepara. Penelitian tersebut didokumentasikan dengan menggunakan *software* Microsoft Excel.
5. Umi Markhamah (2017) berjudul Pengembangan Sistem Informasi Koperasi Simpan Pinjam Berbasis Dekstop dengan melakukan implementasi secara langsung di Koperasi Sari Usaha Tani Jambi. Aplikasi desktop tersebut diintegrasikan dengan basis data sebagai penyimpanan data simpan pinjam di Koperasi Sari Usaha Tani.
6. Itha Aprliai Prihatini (2018) yang berjudul Sistem Penjadwalan Pendadaran atau Ujian Tugas Akhir Berbasis Dekstop yang mengaplikasikannya di Program Studi Teknologi Informasi UMY.

Berdasarkan beberapa tinjauan pustaka di atas, maka penulis membuat aplikasi pengukur *maturity level* menggunakan *framework* COBIT 4.1 menggunakan bahasa pemrograman C# dan berbasis desktop dengan tampilan windows form sebagai pilihan baru dari metode perhitungan sebelumnya yang masih mengukur secara manual menggunakan Microsoft Excel.

Sistem aplikasi berbasis dekstop juga telah banyak digunakan dalam pembuatan aplikasi yang mudah diakses dan digunakan pada berbagai organisasi maupun instansi. Maka, aplikasi pengukur *maturity level* menggunakan *framework* COBIT 4.1 ini dapat dengan baik dan selaras jika dibuat dengan metode dan proses berbasis desktop dengan memiliki beberapa fitur seperti:

1. Aplikasi dapat mengambil data pertanyaan kuisisioner secara otomatis.
2. Aplikasi dapat menghitung hasil perhitungan *maturity level* secara otomatis.
3. Proses pemilihan proses TI, penilaian masing-masing pertanyaan, perhitungan hasil akhir, penyampaian data rekapitulasi, dan pencetakan hasil akhir terdapat pada satu aplikasi.

## 2.2. Landasan Teori

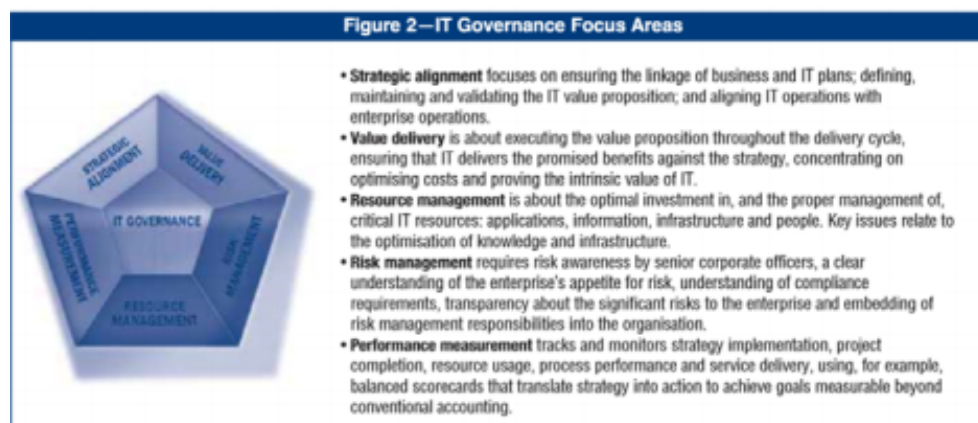
### 2.2.1. Tata Kelola Teknologi Informasi

Pengertian dari Tata Kelola Teknologi Informasi adalah tanggung jawab dewan direktur dan manajemen eksekutif yang terdiri atas kepemimpinan, struktur organisasi dan proses yang memastikan bahwa TI perusahaan mendukung dan memperluas strategi dan tujuan perusahaan (IT ITGI, 2007). Tata Kelola Teknologi Informasi mencakup informasi, teknologi, dan komunikasi, bisnis, dan hukum serta isu yang lainnya yang melibatkan hampir seluruh pemangku kepentingan (*stakeholders*).

Tata kelola teknologi informasi (*IT Governance*) adalah suatu struktur dan proses yang saling terkait serta mengarahkan dan mengendalikan suatu organisasi dalam mencapai tujuan organisasi dengan menyeimbangkan antara resiko dan manfaat dari teknologi informasi serta prosesnya. Maka dari penjelasan di atas dapat disimpulkan bahwa dalam rangka pencapaian tujuan organisasi pengaplikasian tata kelola teknologi informasi adalah pengelolaan yang berkaitan dengan organisasi. (IT ITGI, 2007)

Fokus area tata kelola teknologi informasi dibagi menjadi 5 bagian, yaitu:

*Strategic Alignment, Value Delivery, Resource Management, Risk Management, dan Performance Measurement.* Digambarkan seperti gambar 2.1 di bawah ini :



**Gambar 2.1** *IT Governance Focus Area (IT ITGI, 2007)*

1. **Strategic Alignment:** memastikan hubungan relasi antara bisnis dengan rangkaian rencana teknologi informasi, pemeliharaan, dan validasi nilai-nilai teknologi informasi untuk menyelaraskan tujuan bisnis dan tujuan teknologi teknologi.

2. **Value Delivery:** menjalankan penempatan dari nilai seluruh siklus *delivery*, memastikan bahwa teknologi informasi dapat bermanfaat sesuai dengan tujuan bisnis yang direncanakan dalam strategi, serta perhitungan biaya untuk mengoptimalkan dan membuktikan nilai intrinsik dari teknologi informasi.

3. **Resource Management:** investasi yang efisien dan optimal dalam pengelolaan sumber daya teknologi informasi seperti informasi, aplikasi, SDM, infrastruktur dan pengoptimalisasian infrastruktur.

4. **Risk Management:** mengoptimalkan kesadaran dalam hal pengelolaan risiko oleh petinggi jabatan pada perusahaan, serta bagaimana mendeskripsikan persyaratan kepatuhan, keterbukaan tentang risiko yang signifikan terhadap perusahaan dalam menanamkan tanggung jawab manajemen risiko di sebuah lembaga atau organisasi.

5. **Performance Measurement:** analisa kinerja dan implementasi dari penggunaan sumber daya, penyelesaian proyek, strategi, kinerja proses, dan beberapa tugas lainnya, misalnya : *balanced scorecard* yang menerjemahkan strategi ke dalam tindakan untuk mencapai tujuan yang terukur.

#### 2.2.2. COBIT (Control Objective for Information and Related Technology)

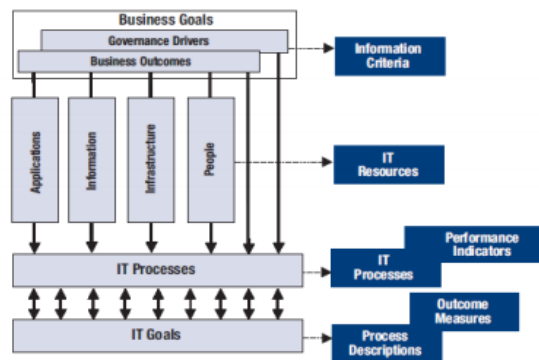
*Control Objective for Information and Related Technology* (COBIT) adalah seperangkat pedoman umum untuk manajemen TI yang dibuat oleh Information System Audit and Control Association (ISACA), dan IT Governance Institute (ITGI) pada tahun 1992 untuk kebutuhan-kebutuhan informasi perusahaan dalam proses mencapai tujuan yang diinginkan, oleh karena itu berikut beberapa prinsip dasar COBIT meliputi (Simonsson & Johnson, 2006):

1. *Business Information Requirement* yaitu berupa pedoman unsur-unsur yang harus ada di dalam sebuah informasi antars lain *effectiveness* (efektif), *efficiency*

(efisien), *confidentiality* (keyakinan), *integrity* (integritas), *availability* (tersedia), *compliance* (pemenuhan), *reliability* (dipercaya).

2. *IT Resource*, terdiri dari pengguna (*people*), aplikasi (*application*), teknologi (*technology*), infrastruktur (*infrastructure*), informasi (*information*), *Database Management System*, *Hardware*, *Software* dan *Multimedia*.

3. *High Level IT Process*, terdiri dari: Proses TI *Planning and Organization* (PO), *Acquisition and Implementation* (AI), *Delivery Support* (DS), dan *Monitoring and Evaluation* (ME).



**Gambar 2.2** COBIT *management, control, alignment and monitoring* (IT ITGI, 2007)

Tujuan utama dari dibuatnya COBIT adalah memberikan kebijaksanaan yang jelas dan latihan yang bagus untuk Tata Kelola Teknologi Informasi, yang digunakan pada organisasi di seluruh dunia dengan harapan dapat membantu proses pengelolaan untuk memahami dan mengatur resiko yang berhubungan dengan Teknologi Informasi. Cara kerja COBIT adalah dengan menyediakan kerangka kerja IT Governance dan petunjuk control obyektif yang rinci bagi manajemen, pemilik proses bisnis, pemakaian dan auditor.

COBIT merupakan sebuah kerangka keseluruhan yang dapat membantu perusahaan dalam mencapai tujuannya pada tata kelola teknologi informasi di sebuah instansi atau perusahaan. Contohnya yaitu COBIT 4.1 membantu perusahaan menciptakan nilai optimal dari teknologi informasi dengan cara

menjaga keselarasan antara mengoptimalkan tingkat resiko penggunaan sumber daya dan pendapatan keuntungan. COBIT 4.1 bersifat umum dan digunakan untuk segala jenis ukuran perusahaan, baik itu sektor non profit, sektor komersial atau pada sektor pemerintahan dan publik.

### **2.2.3. Domain dan Proses TI pada COBIT 4.1**

Aktivitas TI pada COBIT 4.1 didefinisikan pada 4 (empat) proses TI yaitu (IGI ITGI, 2007):

#### **A. *Plan and Organise (PO)***

Domain PO mengutarakan proses yang harus dijalani untuk mengidentifikasi metode agar Teknologi Informasi dapat memberikan kontribusi dalam pencapaian tujuan bisnis organisasi, serta merencanakan, mengkomunikasikan, dan mengelola visi yang ingin dicapai organisasi. Proses TI PO terdiri dari 10 *control objectives*, yaitu:

1. **PO1:** *define a strategic IT plan* (menetapkan rencana strategi TI).
2. **PO2:** *define the information architecture* (menentukan arsitektur informasi).
3. **PO3:** *determine technological direction* (menentukan arah teknologi)
4. **PO4:** *define the IT processes, organization and relationships* (menetapkan proses TI, organisasi dan hubungan).
5. **PO5:** *manage the IT investement* (mengelola investasi TI).
6. **PO6:** *communicate management aims and direction* (mengkomunikasikan tujuan dan arah manajemen).
7. **PO7:** *manage IT human resource* (mengelola sumber daya manusia).
8. **PO8:** *manage quality* (mengelola kualitas)
9. **PO9:** *assess and manage IT risks* (menilai dan mengelola resiko TI)
10. **PO10:** *manage project* (mengelola proyek).

## **B. *Acquire and Implement (AI)***

Domain AI berisi tentang proses-proses yang dilakukan dengan cara mengidentifikasi, membangun atau menyediakan aplikasi Teknologi Informasi untuk mewujudkan rencana Teknologi Informasi. Selain itu, perancangan yang dilakukan dan pemeliharaan terhadap sistem TI juga menjadi cakupan domain ini. Proses TI *AI* terdiri dari 7 *control objectives*, yaitu:

1. **AI1:** *identify automated solutions* (identifikasi solusi otomatis)
2. **AI2:** *acquire and maintain application software* (memperoleh dan memelihara aplikasi perangkat lunak)
3. **AI3:** *acquire and maintain technology infrastructure* (memperoleh dan memelihara infrastruktur teknologi)
4. **AI4:** *enable operation and use* (pengaktifan operasi dan penggunaan)
5. **AI5:** *procure IT resource* (pengadaan sumberdaya TI)
6. **AI6:** *manage changes* (mengelola perubahan)
7. **AI7:** *install and accredit solutions and changes* (install dan akreditasi solusi dan perubahan).

## **C. *Deliver and Support (DS)***

Domain DS fokus pada memberikan dukungan agar pencapaian hasil sistem Teknologi Informasi sesuai dengan yang diharapkan. Proses ini secara garis besar terdiri dari keamanan, aspek kontinuitas, sampai dengan memberikan pelatihan kepada pengguna. Proses TI *DS* terdiri dari 13 *control objective*, yaitu:

1. **DS1:** *define and manage service levels* (mendefinisikan dan mengelola tingkat layanan)
2. **DS2:** *manage third-party service* (mengelola layanan pihak ketiga)
3. **DS3:** *manage performance and capacity* (mengelola kinerja dan kapasitas)
4. **DS4:** *ensure continuous service* (memastikan layanan secara terus menerus)
5. **DS5:** *ensure system security* (memastikan sistem keamanan)

6. **DS6:** *identify and allocate costs* (mengidentifikasi dan mengalokasikan biaya)
7. **DS7:** *educate and train users* (mendidik dan melatih pengguna)
8. **DS8:** *manage service desk and incidents* (mengelola pelayanan meja dan insiden)
9. **DS9:** *manage the configuration* (mengelola konfigurasi)
10. **DS10:** *manage problems* (mengelola masalah)
11. **DS11:** *manage data* (mengelola data)
12. **DS12:** *manage the physical environment* (mengelola lingkungan fisik)
13. **DS13:** *manage operations* (mengelola operasi).

#### **D. Monitor and Evaluate (ME)**

Proses TI perlu dinilai secara berkala dari waktu ke waktu untuk kualitas dan pemenuhan persyaratan dan ini berlaku untuk semua Proses TI. Proses TI membahas pemantauan pengendalian internal, manajemen kerja, kepatuhan terhadap tata kelola yang telah diatur. Proses TI ME terdiri dari 4 *control objective*, yaitu:

1. **ME1:** *monitor and evaluate IT performance* (memonitor dan mengevaluasi kinerja TI)
2. **ME2:** *monitor and evaluate internal control* (memonitor dan mengevaluasi pengendalian internal)
3. **ME3:** *ensure compliance with external requirements* (memastikan kepatuhan terhadap persyaratan eksternal)
4. **ME4:** *provide IT Governance* (menyediakan pengelolaan TI).



#### 2.2.4. COBIT 4.1 *Maturity Level*

(IGI ITGI, 2007) *Maturity Model* adalah sebuah metode yang digunakan untuk mengukur sejauh mana kapabilitas manajemen dalam tingkat pengembangan manajemen proses. Tercapainya tujuan-tujuan COBIT yang sudah diterapkan berguna untuk melihat bagus atau tidaknya suatu pengembangan atau kapabilitas manajemen. Salah satu alat perhitungan kinerja di suatu sistem teknologi informasi adalah level kematangan (*Maturity Level*).

Untuk menentukan *maturity level* dalam pengelolaan dan pengendalian pada proses teknologi informasi harus didasarkan pada metode evaluasi organisasi sehingga dapat mengevaluasi sendiri dari level 0 (tidak ada) hingga level 5 (optimal). Penggunaan *maturity level* yang dikembangkan untuk setiap 34 proses teknologi informasi memungkinkan manajemen dapat mengidentifikasi:

1. Kondisi instansi atau perusahaan saat ini
2. Kondisi industri sekarang untuk perbandingan
3. Kondisi instansi atau perusahaan dari standar internasional sebagai perbandingan tambahan
4. Kondisi yang diinginkan instansi atau perusahaan.



Gambar 2.3 Grafik *Maturity Model*

Berikut ini merupakan tabel pengelompokan berdasarkan nilai *maturity level* yang telah terinci:

**Tabel 2.1** Index *Maturity Level*

<i>Maturity Indexes</i>	<i>Maturity Level</i>
<b>0-49</b>	<b>0-Non Existent</b>
<b>0.50-1.49</b>	<b>1-Initial / Ad Hoc</b>
<b>1.50-2.49</b>	<b>2-Repeateble But Intuitive</b>
<b>2.50-3.49</b>	<b>3-Defined Process</b>
<b>3.50-4.49</b>	<b>4-Managed And Measurable</b>
<b>4.50-5.00</b>	<b>5-Optimized</b>

Penjelasan setiap *maturity level* adalah sebagai berikut:

1. **Non-Exsistent** (0 = *Management process are not applied at all*), pernyataan minimal dan menyeluruh terhadap semua proses yang dapat dikenali. Bahkan perusahaan tidak mengetahui bahwa terhadap permasalahan yang harus diatasi dan solusi yang harus diambil.
2. **Ad-Hoc** (1 = *Processes are ad hoc and disorganized*), Terdapat bukti bahwa perusahaan mengetahui adanya permasalahan yang harus diatasi. Bagaimanapun juga tidak terdapat proses standar, namun menggunakan pendekatan *ad hoc* yang cenderung diperlakukan secara individu atau per kasus. Secara umum pendekatan kepada pengelolaan proses tidak terorganisasi.
3. **Reapetable** (2 = *Process or alow a regular pattern*), Proses dikembangkan kedalam tahapan dimana prosedur serupa diikuti oleh pihak-pihak yang berbeda untuk pekerjaan yang sama. Tidak terdapat pelatihan formal atau pengkomunikasian prosedur standard an tanggung jawab diserahkan kepada individu masing-masing.
4. **Defined** (3 = *Processes are documented and communicated*), Prosedur distandarisasi dan didokumentasikan kemudian dikomunikasikan melalui pelatihan. Kemudian diamanatkan bahwa proses-proses tersebut harus diikuti. Namun penyimpangan tidak mungkin dapat terdeteksi. Prosedur sendiri tidak lengkap namun sudah menformalkan praktek yang berjalan.

5. **Managed** (4 = *Processes are monitored and measured*), Manajemen telah mengukur dan mengawasi ketaatan terhadap prosedur dan mengambil tindakan jika proses tidak dapat dikerjakan secara efektif, proses ini berada dibawah penyediaan praktek yang baik dan peningkatan yang konstan. Otomatisasi dan perangkat digunakan dalam batasan tertentu.
6. **Optimized** (5 = *Best practices are followed and automated*), Proses telah dipilih ke dalam tingkat praktek yang baik, berdasarkan hasil dari perbaikan berkelanjutan dan permodelan kedewasaan dengan perusahaan lain. Teknologi Informasi digunakan sebagai proses integrasi untuk mengotomatisasi alur kerja, penyediaan alat untuk peningkatan kualitas dan efektifitas, serta mewujudkan sebuah perusahaan yang dapat beradaptasi.

#### 2.2.5. Perhitungan *Maturity Level*

Perhitungan level kematangan pada setiap proses Ti di dalam COBIT 4.1 menggunakan skala penilaian yang tertera pada tabel 2.2 (Laksito & Luthfi, 2013):

**Tabel 2.2** Skala Penilaian

<i>Value Index</i>	<b>Statements Compliance Values</b>
Tidak benar sama sekali	0
Ada benarnya	0,33
Sebagian besar benar	0,66
Sepenuhnya benar	1

Rumusan perhitungan total nilai *maturity level* di dalam COBIT 4.1 adalah dengan cara seperti pada gambar 2.4.

$$\begin{array}{l}
 \text{Maturity level compliance value (a)} \\
 = \frac{\text{Sum of statements compliance values}}{\text{Number of Maturity level statements}} \\
 \\
 \text{Normalized compliance value (b)} \\
 = \frac{\text{(a)}}{\text{Sum of Maturity level compliance value}}
 \end{array}$$

**Gambar 2.4** Rumusan Perhitungan *Maturity Level*

## **2.3. Teknologi Pengembangan Aplikasi**

### **2.3.1. Unified Markup Language (UML)**

*Unified Markup Language* (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk mespesifikasikan, memvisualkan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak, menurut (Booch & Ivar, n.d.). UML adalah bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun dokumen artifak dari software.

Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case* Diagram, ER Diagram, dan *Class* Diagram.

#### **a. *Use Case* Diagram**

*Use Case* merupakan salah satu UML yang digunakan dalam membuat desain sebuah aplikasi. *Use Case* menjelaskan interaksi yang terjadi antara “aktor” yang mana inisiator dari interaksi sistem itu sendiri dengan sistem yang ada (Brady & Loonam, 2010). *Use Case* memiliki beberapa komponen yaitu:

1. Aktor

Menggambarkan seseorang yang berinteraksi dengan sistem, di mana hanya dapat meng-*input*-kan informasi dan menerima informasi dari sistem dan tidak memegang kendali pada *Use Case*.

2. *Use Case*

Gambaran fungsional sistem yang akan dibuat, agar pengguna lebih mengerti penggunaan sistem.

**b. Entity Relationship Diagram**

*Entity Relationship Diagram* (ERD) adalah suatu Teknik yang digunakan untuk memodelkan diagram untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki relasi, biasanya oleh *system analysis* dalam tahap analisis persyaratan proyek pengembangan sistem (Brady & Loonam, 2010). ERD memiliki tiga komponen yaitu:

1. Entitas

Entitas memiliki simbol yang biasanya digambarkan dengan persegi panjang. Entitas merupakan objek yang mewakili elemen yang nyata dan dapat dibedakan dari satu sama lain.

2. Atribut

Setiap entitas memiliki elemen yang disebut dengan atribut yang berfungsi untuk mendeskripsikan karakteristik pada entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi elemen satu dengan yang lain.

3. Relasi

Relasi merupakan hubungan antara kumpulan dari sebuah entitas yang berasal dari himpunan entitas yang berbeda.

**c. Class Diagram**

*Class Diagram* adalah salah satu model statis yang menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*. *Class diagram* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas dan hubungan mereka (“4\_Class Diagram.pdf,” n.d.). *Class diagram* mirip dengan ER Diagram pada perancangan database, bedanya pada ER diagram tidak terdapat operasi atau *method* tapi hanya atribut. *Class diagram* memiliki empat elemen yaitu *Class*, *Attributes*, *Operations*, dan *Relationships*. (“The class diagram,” 2004).

### 2.3.2. SQL Server

Microsoft SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur *client server*. Fitur pada Microsoft SQL Server yaitu mempunyai kemampuan untuk membuat basis data mirroring dan clustering. Microsoft SQL Server juga mendukung SQL sebagai bahasa untuk memproses *query* ke dalam basis data dan kita tahu bahwa SQL ini sudah digunakan secara umum pada semua produk *database server* (Delano & George, 2005).

### 2.3.3. Bahasa Pemrograman C#

C# adalah sebuah bahasa pemrograman yang diciptakan oleh microsoft dibawah kepemimpinan Andres Hejlsberg yang sebelumnya juga telah menciptakan berbagai macam bahasa pemrograman termasuk C++, Borland Turbo dan Borland Dephi. Bahasa C# juga telah menjadi standar yang diakui internasional oleh ECMA. C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis Windows (desktop) dan aplikasi berbasis *website* serta aplikasi berbasis web services seperti halnya pemrograman yang lain, (Enterprise, 2016).

C# didesain untuk membangun jangkauan aplikasi *enterprise* yang berjalan di atas framework .NET.