



Hak cipta dan penggunaan kembali:

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

Copyright and reuse:

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

BAB II

TINJAUAN PUSTAKA

Berikut adalah tinjauan pustaka berupa landasan teori yang digunakan dalam penelitian ini.

2.1 Sistem Informasi Penjualan

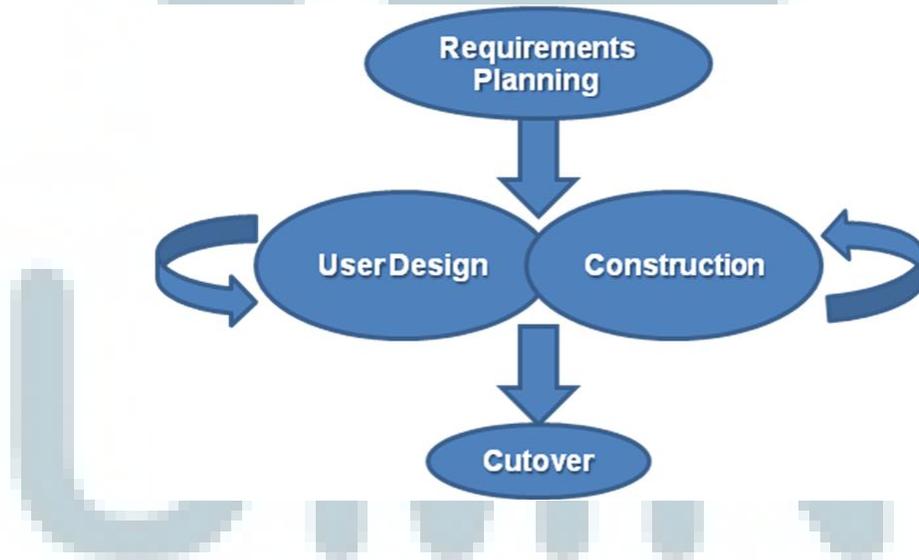
Sistem informasi penjualan adalah perangkat fleksibel untuk melakukan pengumpulan, konsolidasi, dan memanfaatkan data dari proses penjualan dan distribusi. Sistem informasi penjualan dapat digunakan pada untuk proses pembuatan keputusan sebagai instrumen pengontrolan dan perencanaan sehingga dapat mengikuti perkembangan pasar dan tren ekonomi lebih cepat untuk diikut. (SAP, 2018)

Sistem Informasi adalah komponen yang saling berhubungan untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan control dalam suatu organisasi. Sistem informasi penjualan juga membantu *manager* untuk menganalisa permasalahan, menggambarkan objek yang kompleks serta membuat suatu produk. (Hartono, Rostianingsih, & Setiabudi, 2015)

Jika pengertian dari kedua teori tersebut digabung maka pengertian sistem informasi penjualan adalah pembuatan pernyataan penjualan melalui serangkaian prosedur-prosedur dengan menggunakan teknologi informasi untuk menyimpan dan menampilkan data.

2.2 Rapid Application Development

Merupakan suatu jenis dari *System Development Life Cycle* atau siklus hidup daripada sebuah sistem. Siklus ini merupakan salah satu siklus pengembangan sistem yang singkat perputarannya. Siklus *Rapid Application Development* atau dikenal juga dengan RAD menurut James Martin terdiri dari 4 buah fase, yaitu fase *Requirement Planning*, *User Design & Construction*, dan *Cutover*. (Naz & Khan, 2015)



Gambar 2. 1 Rapid Application Development

Fase *Requirement Planning* merupakan fase untuk mengumpulkan dan menyusun suatu kebutuhan user terhadap sistem tersebut. Fase *User Design & Construction* adalah fase untuk melakukan perancangan dan pembangunan sistem sesuai dengan kebutuhan yang telah direncanakan. Fase *Cutover* adalah fase setelah berjalannya pembangunan dan melakukan implementasi sistem yang telah dibangun.

Pendekatan *Rapid Application Development* ini mengimprovisasikan dan memudahkan desain dari arsitektur yang dapat dikonfigurasi kembali karena model arsitektur yang sudah ditetapkan terdapat didalamnya. (Naz & Khan, 2015).

Rapid Application Development masuk kedalam metode pengembangan sistem jenis *Agile* yang mana melibatkan komunikasi, perencanaan, *testing*, dan integrasi yang permanen. (Stoica, Mircea, & Ghilic-Micu, 2013)

Berdasarkan pemahaman yang telah disampaikan oleh para ahli, dapat dimengerti bahwa metode *Rapid Application Development* adalah siklus pengembangan sistem yang singkat untuk mengembangkan sistem atau aplikasi secara cepat, dan membutuhkan umpan balik dari *user* dalam pengembangannya.

2.3 Waterfall

Model *Waterfall* merupakan model SDLC tertua dan yang paling banyak diketahui. Model ini banyak digunakan pada proyek pemerintah dan banyak perusahaan besar. (Alshamrani & Bahattab, 2015)

SDLC model *Waterfall* adalah metode pengembangan *software* secara sekuensial yang mana prosesnya bertambah secara menurun seperti air terjun melalui daftar dari fase-fase yang harus dilakukan secara urut untuk sukses dalam membuat *software*. Model *Waterfall* memiliki lima fase yaitu: *analysis, design, implementation, testing, dan maintenance*. (Bassil, 2012)

Berdasarkan teori yang dipaparkan oleh para ahli, maka dapat disimpulkan model *Waterfall* merupakan model pengembangan sistem yang dilakukan secara urut dan menurun sesuai dengan fase-fase yang ada. Jika ada perubahan ditengah tahap pembangunan maka perubahan tersebut bisa dilakukan jika siklus telah selesai.

2.4 Database

Database atau basis data adalah satu atau sekumpulan tempat penyimpanan untuk data yang dapat digunakan secara simultan oleh banyak departemen dan banyak pengguna (Connelly T, 2015).

Penggunaan sistem basis data dilandasi dengan keinginan untuk melakukan integrase terhadap data operasional dari perusahaan untuk

menyediakan data yang tersentralisasi dengan akses yang terkontrol (Özsu & Valduriez, 2011)

Setelah dijelaskan oleh para ahli, maka dapat diambil kesimpulan yaitu *database* merupakan tempat penyimpanan data yang terorganisir untuk bisa digunakan secara mudah dan simultan.

2.5 Object-oriented analysis and design

OOAD merupakan proses teknis yang diterima secara global untuk memanipulasi aplikasi, bisnis atau model sistem, dan diagram grafis untuk melakukan analisa dan improvisasi kualitas produk dengan mengaplikasikan metode *prototype* berorientasi objek. (Mukherjee, 2016)

Menurut Mathiassen, Munk-Madsen, Nielsen, dan Strage pada buku *Object Oriented Analysis and Design*, keuntungan dalam menggunakan analisa dan perancangan sistem berorientasi adalah:

1. Dapat digunakan untuk memodelkan hampir semua fenomena.
2. Memberi informasi yang jelas tentang konteks dari sistem.
3. Mengurangi biaya *maintenance*
4. Memudahkan untuk mencari hal yang akan diubah karena dokumentasi yang jelas.

5. Membuat perubahan sistem menjadi local sehingga tidak mempengaruhi modul yang lain.

Analisa dan perancangan dengan orientasi obyek dapat disimpulkan sebagai cara untuk membangun sistem dengan lebih mudah, dan lebih kuat karena keuntungannya dan perlakuannya terhadap obyek.

2.6 C#

Merupakan Bahasa pemrograman yang umumnya dipergunakan untuk program yang berbasis *platform* Windows. C# merupakan Bahasa pemrograman yang sudah berorientasi kepada pemrograman objek (*Object oriented programming*) yang dikembangkan oleh Microsoft bersama dengan *platform* .NET (Nakov, 2013) Bahasa ini menjadi sangat populer untuk developer program karena bahasa yang digunakan mudah dipahami untuk para pemula, namun tetap digunakan dalam banyak industri. (Nakov, 2013)

Dari teori yang didapatkan, maka bahasa C# merupakan bahasa pemrograman obyek yang digunakan untuk mengembangkan aplikasi untuk *platform* Windows keluaran Microsoft.

2.7 Entity Framework

Entity Framework adalah *Framework* .NET yang memungkinkan *developer* untuk menggunakan hubungan data dengan menggunakan objek

domain yang spesifik. (Microsoft, Introduction to Entity Framework, 2016). Dengan menggunakan *Entity Framework*, maka kode yang biasa ditulis dalam program untuk melakukan akses terhadap data tidak diperlukan lagi, berbeda dengan *framework* *linq to sql* yang harus menulis kode dalam program agar bisa terhubung dengan sistem basis datanya.

Keuntungan dalam menggunakan *framework* ini adalah tidak diperlukan kode untuk database dalam proses pembuatan sistem jika menggunakan metode *code first* sedangkan jika menggunakan metode *database first* maka *entity* dalam program akan otomatis disediakan oleh *framework* tersebut sehingga memungkinkan untuk membuat program lebih efisien dalam pemrosesan data. (Microsoft, Microsoft docs, 2010)

2.8 LINQ to SQL

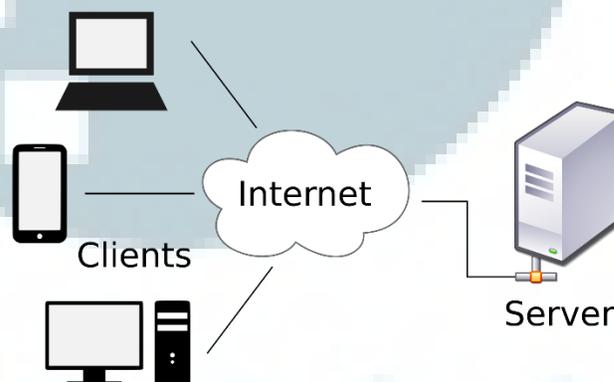
LINQ atau *Language Integrated Query* adalah model pemrograman yang digunakan untuk menambahkan fitur *query* pada Bahasa pemrograman berbasis .NET dari *Microsoft* yang diperkenalkan pada *visual studio 2008* dan *.NET Framework 3.5*. Pengembang dapat menuliskan *query* kedalam kode untuk mendapatkan data dari berbagai sumber data. (O Khanna & A Mistry, 2012)

LINQ to SQL adalah bagian dari *ADO.NET* yang menerjemahkan LINQ pada model objek kepada SQL dan mengirimkannya kepada sistem basis data untuk dieksekusi. Saat sistem basis data mengembalikan hasilnya, *LINQ to SQL*

akan menerjemahkan kembali kepada objek agar dapat berfungsi. (O Khanna & A Mistry, 2012)

2.9 Client Server Architecture

Client-Server adalah model dalam jaringan pengembangan dari *file server*. Model dari arsitektur ini terdapat 2 buah jenis yaitu *client* dan *server*, yang mana keduanya memiliki fungsi yang berbeda. *Server* berperan sebagai penyedia data, aplikasi, fungsi daripada *hard drive*, *modem*, *printer*, dan sebagainya.



Gambar 2. 2 Gambar arsitektur *client server*.

Server akan menunggu permintaan dari *client* untuk melakukan pengolahan data dan memberikan hasilnya kepada *client*. *Client* berperan untuk mengirimkan permintaan kepada *server* atas suatu aktivitas yang mana akan

diproses oleh *server* dan menunggu hasil dari *server* yang akan ditampilkan kepada *client* (Oluwatosin, 2014).

Client-Server membuat komunikasi intra proses karena melibatkan pertukaran data antara *client* dan *server* dimana masing-masing menjalankan fungsi yang berbeda. (Oluwatosin, 2014)

Arsitektur ini menggunakan protocol *Transmission Control Protocol/Internet Protocol* atau disebut juga dengan TCP/IP untuk melakukan komunikasi antara *client* dengan *server*.

UMMN