

## **Bab 2**

### **Tinjauan Pustaka**

#### **2.1. Sistem**

Menurut Gary B. Shelly dan Harry J. Rosenblatt [1], Sistem adalah sekumpulan komponen terkait yang menghasilkan hasil tertentu. Andalia, Fanny dkk. [2] dalam jurnalnya berpendapat bahwa sistem merupakan elemen-elemen yang saling berkaitan. untuk mencapai tujuan tertentu. Sistem harus memiliki organisasi hubungan antar elemen yang baik. Dapat disimpulkan bahwa sistem adalah sekumpulan elemen yang terstruktur, saling berhubungan dan berinteraksi untuk mencapai tujuan bersama.

##### **2.1.1 Pengertian Informasi**

Menurut Gary B. Shelly dan Harry J. Rosenblatt informasi adalah data yang telah diubah menjadi keluaran yang berguna terhadap aktifitas pengambilan keputusan [1]. Data adalah fakta yang jelas lingkup, tempat dan waktu-nya, data juga dapat dikatakan bahan mentah. Sedangkan informasi bahan yang siap digunakan [2] . Informasi berdasarkan manajerialnya dibagi menjadi tiga yaitu: [3]

1. informasi strategis
2. informasi taktis
3. informasi operasional

Informasi merupakan data yang telah diolah untuk mendapatkan arti dan memperbaiki proses pengambilan keputusan. Ada beberapa karakteristik idata yang berkualitas [1], yaitu:

1. Akurat: Informasi yang berkualitas merupakan data yang tepat,, benar, konsisten dan dapat digunakan. Data yang akurat dapat mempengaruhi hasil dari pengambilan keputusan

2. Tepat waktu: Informasi harus tepat waktu, dimana ketersediaan data pada waktu yang tepat agar penggunaan data atau informasi yang telah didapat dapat digunakan dengan tepat.
3. Relevan: Data yang digunakan harus relevan dalam konteks maupun subjeknya.
4. Cukup: Data yang cukup juga dapat mendukung kualitas informasi yang didapat.
5. Sebanding dengan biaya: Penggunaan data harus bijak agar dapat mengimbangi biaya yang digunakan dalam memproses data.

### **2.1.2 Pengertian Sistem Informasi**

Sistem informasi merupakan suatu sistem didalam sebuah organisasi yang dapat mendukung operasional organisasi, merupakan kegiatan manajerial dan merupakan strategis suatu organisasi yang dapat menyediakan laporan yang diperlukan [4] Menurut A. Agung sistem informasi adalah kumpulan elemen yang berkaitan dan saling mempengaruhi dalam melakukan kegiatan dengan tujuan bersama [5].

Dapat disimpulkan bahwa sistem informasi merupakan suatu kombinasi elemen yang berasal dari komponen yang berkaitan dengan *hardware, software, people* dan *network* dan menghasilkan informasi untuk mencapai tujuan.

## **2.2 Microsoft**

*Microsoft Corporation* pertama kali dirintis oleh seorang mahasiswa bernama Bill Gates yang didirikan pada tahun 1975. *Microsoft* dikenal sebagai salah satu perusahaan yang mengembangkan sistem operasi komputer. Sebagai perusahaan yang bergerak di bidang pengembangan teknologi, *Microsoft* memiliki tujuan bisnis yaitu, mengembangkan *Windows Client*, yaitu sistem operasi berbasis *Microsoft Windows* pada *client* dan *server*,

### **2.2.1 Microsoft Windows**

*Microsoft Windows* merupakan salah satu sistem operasi computer yang paling banyak digunakan di seluruh dunia. *Microsoft windows* mulai dikembangkan berdasarkan data berbasis DOS. DOS (*Disk Operating System*) merupakan salah satu

sistem operasi komputer sistem operasi yang beroperasi menggunakan disk didalam komputer.

Awal sejarah *Microsoft* berawal dari Paul Allen dan Bill Gates, mencoba untuk membuat bisnis yang sukses memanfaatkan kemampuan pemrograman mereka. Majalah edisi 1975 Januari *Popular Electronics* menampilkan *Altair 8800* mikrokomputer buatan *Micro Instrumentation dan Telemetry Systems (MITS)*, mengetahui itu mereka membuat penerjemah untuk MITS namun menggunakan simulator untuk *Altair* dan juga mengembangkan interpreturnya. Meskipun simulator, sistem tersebut beroperasi sangat baik dan MITS setuju untuk mendistribusikannya dengan nama *Altair Basic*. Pada 4 April 1975, *Microsoft* didirikan dan dibesarkan Bill Gates sebagai CEOnya.

### **2.2.2 Microsoft access**

*Microsoft access* merupakan aplikasi yang dapat mengolah *database*, biasanya digunakan untuk rnengolah beberapa jenis data dengan pengoperasian yang mudah. *Microsoft access* tidak memiliki fasilitas kompilasi, oleh karena itu terasuk kedalam kelompok interpreter. Program *Microsoft access* adalah program yang bekerja *under windows*, jika dibandingkan dengan proqram yang ada, *Microsoft Access* tergolong lebih baru dibandingkan program yang ada sebelumnya dalam *under dos* [6].

### **2.3 Database**

*Database* adalah kumpulan data yang diorganisasikan untuk melayani banyak aplikasi secara efisien dengan memusatkan data dan mengendalikan data yang berlebihan [4]. *Database* juga bisa diartikan sebagai berikut [7] ;

1. Gabungan *file* yang saling berelasi, relasi tersebut ditunjukkan dengan kunci dari tiap *file* yang ada untuk digunakan dalam satu lingkup perusahaan, instansi
2. Kumpulan *file* data yang terorganisasi, terintegrasi, dan bisa dipakai bersama.
3. Kumpulan rekaman data berbagai tipe yang memiliki relasi satu sama lain
4. Sekumpulan data organisasi untuk melayani banyak aplikasi secara efisien dengan memusatkan data dan mengendalikan redundansi data.

5. Kumpulan dari data yang saling terintegrasi satu dengan yang lainnya tersimpan dalam perangkat keras komputer dan menggunakan perangkat lunak untuk bantuan dalam mengoperasikannya.

### **2.3.1 Sistem *database* manajemen**

Sistem manajemen basis data (DBMS) adalah perangkat lunak yang memungkinkan organisasi untuk memusatkan data, mengelolanya secara efisien, dan menyediakan akses ke data yang disimpan oleh program aplikasi. DBMS bertindak sebagai antarmuka antara program aplikasi dan file data fisik. Ketika program aplikasi memanggil item data, seperti gaji kotor, DBMS menemukan item ini dalam *database* dan menyajikannya ke program aplikasi. Menggunakan file data tradisional, programmer harus menentukan ukuran dan format setiap elemen data yang digunakan dalam program dan kemudian memberitahu komputer di mana mereka berada [4].

### **2.3.2. *Database Management System* (DBMS) relasional**

DBMS kontemporer menggunakan model *database* yang berbeda untuk melacak entitas, atribut, dan hubungan. Jenis DBMS yang paling populer saat ini untuk PC serta komputer yang lebih besar dan mainframe adalah DBMS relasional. *Database* relasional mewakili data sebagai tabel dua dimensi (disebut relasi). Tabel dapat disebut sebagai *file*. Setiap tabel berisi data tentang entitas dan atributnya. *Microsoft Access* adalah DBMS relasional untuk sistem desktop, sedangkan *DB2*, *Oracle Database*, dan *Microsoft SQL Server* adalah DBMS relasional untuk mainframe besar dan komputer kelas menengah. *MySQL* adalah DBMS open-source yang populer, dan *Oracle Database Lite* adalah DBMS untuk perangkat komputasi genggam kecil, ERD

Dalam sistem manajemen basis data (DBMS), semua tabel dihubungkan oleh bidang umum. Bidang umum yang umum mungkin berupa Nomor Pelanggan, yang dapat digunakan untuk menemukan informasi tentang pelanggan tersebut di tabel lain. Bidang umum yang menghubungkan dua tabel dikatakan menautkan, menggabungkan, atau menghubungkan tabel. Dalam DBMS, tabel tertaut

membentuk struktur data terpadu yang sangat meningkatkan kualitas dan akses data. Desain ini, juga disebut database relasional atau model relasional, diperkenalkan pada 1970-an dan terus menjadi pendekatan paling populer untuk mengatur, menyimpan, dan mengelola data bisnis.

## **2.4. Diagram**

Diagram merupakan gambar sederhana yang menunjukkan penampilan, struktur, atau cara kerja sesuatu representasi skema. Dalam sistem informasi terdapat beberapa diagram yang digunakan,

### **2.4.1 Data Flow Diagram (DFD)**

Diagram aliran data merupakan teknik analisis yang digunakan untuk memodelkan sistem dari perspektif data yang mengalir di seluruh sistem itu sendiri. DFD dapat digunakan untuk memodernisasi sistem lama dan sama pentingnya dalam elisitasi kebutuhan. Dalam rekayasa perangkat lunak, DFD dapat digunakan untuk memodelkan struktur sistem perangkat lunak. DFD dapat membantu menghasilkan rencana proyek yang jauh lebih akurat. Model DFD secara dasar memfasilitasi pengorganisasian proyek secara logis, serta mengembangkan pemahaman tentang masalah yang ingin diatasi.

DFD dapat berfungsi sebagai teknik pemodelan yang mampu menganalisis kebutuhan pelanggan dan mengubahnya menjadi struktur yang jelas, terutama digunakan dalam pengembangan perangkat lunak untuk menganalisis persyaratan fungsional dan menetapkan desain penuh sistem yang sedang dikembangkan. Jadi, komponen yang berbeda dari sistem dapat diidentifikasi pada berbagai tingkat abstraksi. DFD membantu untuk memperkirakan durasi proyek lebih akurat karena mengidentifikasi seluruh komponen fungsional, sehingga memudahkan untuk memperkirakan tugas untuk masing-masing komponen.

### 2.4.2 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram (ERD)* adalah langkah pertama dalam desain database; ini merupakan langkah penting bagi perancang basis data, pengguna, analis, manajer, dan rekayasa perangkat lunak. *Entity Relationship Diagram (ERD)* terdiri dari kumpulan entitas, hubungan entitas, dan atribut. Entitas merupakan objek yang digunakan untuk kita menyimpan data.

Langkah pertama adalah membuat daftar entitas yang akan diidentifikasi selama fase analisis sistem dan untuk mempertimbangkan sifat hubungan yang menghubungkan. Pada tahap ini dapat menggunakan metode yang disederhanakan untuk menunjukkan hubungan antar entitas. Meskipun ada berbagai cara untuk menggambar ERD, metode yang populer adalah untuk mewakili entitas sebagai persegi Panjang dan hubungan sebagai bentuk berlian. Entitas persegi panjang diberi label dengan kata benda tunggal, dan berlian hubungan diberi label dengan kata kerja, biasanya dengan cara dari atas ke bawah dan dari kiri ke kanan.

### 2.5. Software Development Life Cycle (SDLC)

*Software Development Life Cycle* adalah model yang menggambarkan keseluruhan area tentang bagaimana pengembangan perangkat lunak berlangsung dengan setiap fase menggambarkan cara kerjanya. SDLC adalah pendekatan sistematis untuk menyelesaikan proses pengembangan perangkat lunak dalam waktu dan menjaga kualitas perangkat lunak. Siklus hidup pengembangan sistem menyediakan serangkaian kegiatan yang akan dilakukan selama pengembangan sistem dan sering disebut siklus hidup pengembangan perangkat lunak[8]. Berikut gambar tahapan SDLC yang dapat dilihat pada gambar 2.1.



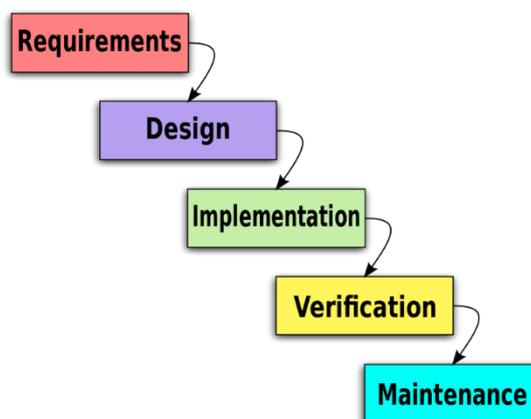
Gambar 2.1 SDLC

### 2.5.1 Model *Waterfall*

Pelopori pengembangan perangkat lunak, Winston Royce pada tahun 1970, mengusulkan air terjun. Model ini adalah salah satu model SDLC tertua, tetapi tidak banyak digunakan dalam beberapa tahun terakhir. Ini mengikuti aliran sekuensial linier di mana kemajuan terlihat mengalir ke bawah saat dalam fase berkembang. Di sini semua persyaratan harus dikumpulkan pada awal memulai proyek dan kemudian pindah ke fase lain. Setiap fase bergantung pada Informasi yang dikumpulkan pada fase sebelumnya karena tidak memungkinkan untuk berpindah ke fase berikutnya hingga fase sebelumnya telah selesai. Pendekatan air terjun tidak memungkinkan proses untuk kembali ke fase sebelumnya dan memungkinkan perubahan di dalamnya. Model *Waterfall* digunakan untuk proyek kecil, karena hanya ada sedikit ruang untuk revisi setelah tahap selesai. Fase-fase dalam model air terjun meliputi : [9].

1. Kebutuhan
2. Desain
3. Implementasi
4. Verifikasi
5. Pemeliharaan

Gambar dari tahapan metode *waterfall* dapat dilihat pada gambar 2.2.



Gambar 2.2 Tahapan model *waterfall*

Jika tahap satu belum selesai, maka tahap dua tidak dapat berjalan, begitupun seterusnya. Setiap tahap saling terkait dan tiap tahapnya harus dikerjakan secara detail. Spesifikasi yang ada pada model *Waterfall* harus didefinisikan dengan rinci pada tahap setiap tahapnya, karena metode *waterfall* sulit melakukan perubahan saat dipertengahan proses pengembangan. Jadi apa yang telah disepakati bersama oleh tim dan klien di tahap awal, itulah yang akan menjadi hasil akhir.

#### 1. Analisis kebutuhan

Analisis kebutuhan merupakan tahap pengambilan data terkait dengan sistem informasi yang akan dibuat. Pengambilan data biasanya dilakukan dengan wawancara, observasi atau penelitian langsung. Tujuan dari tahap ini adalah melakukan investigasi pendahuluan untuk mengevaluasi peluang atau masalah bisnis terkait. Investigasi awal merupakan langkah kritis karena hasilnya akan mempengaruhi keseluruhan proses pengembangan.

#### 2. Sistem desain

Proses ini akan berfokus pada pembangunan struktur data, arsitektur perangkat lunak, perancangan *interface*, perancangan fungsi internal dan eksternal serta detail dari setiap algoritma prosedural. Sistem desain membantu dalam menentukan perangkat keras dan persyaratan sistem dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

#### 3. Implementasi

Implementasi merupakan tahap dimana aplikasi dibuat oleh programmer. Proses pengodingan aplikasi mengacu pada hasil dari tahapan-tahapan sebelumnya.

#### 4. Verifikasi

Verifikasi merupakan tahap dimana aplikasi yang telah dibuat diuji untuk mengetahui sejauh mana aplikasi ini dapat membantu klien dan kelayakannya. Pengerjaan modul yang berbeda dari programmer akan disatukan dan diuji untuk dapat melihat bagaimana aplikasi ini bekerja dan jika ada permasalahan maka akan diperbaiki.

## 5. Perawatan

Tahapan ini pada dasarnya meliputi tahapan penginstalasian *software* dan pengujian aplikasi. *Perawatan* meliputi pemeliharaan, peeningkatan, dan melindungi sistem. *Perawatan* kontrol keamanan melindungi sistem dari ancaman eksternal dan internal. Sistem yang dirancang dengan baik harus aman, andal, dapat dipelihara, dan dapat diskalakan. Desain yang dapat diskalakan dapat diperluas untuk memenuhi persyaratan dan volume bisnis baru. Pengembangan sistem informasi selalu dalam proses. Proses bisnis berubah dengan cepat, dan sebagian besar sistem informasi perlu diperbarui secara signifikan atau diganti setelah beberapa tahun beroperasi.

### 2.5.2 Model iteratif

Model iteratif menggunakan iteratif, merupakan model yang dapat mengatasi kelemahan model air terjun. Berbeda dengan model air terjun di mana persyaratan diperlukan sekali, persyaratan model berulang dikumpulkan di setiap fase. Proyek ini dibagi menjadi komponen-komponen yang lebih kecil sehingga hasilnya dapat dimanfaatkan pada tahap berikutnya. Setelah setiap kenaikan umpan balik dikumpulkan dari klien berdasarkan proses selanjutnya yang direncanakan dan modifikasi dilakukan. Versi baru dari perangkat lunak diproduksi di setiap fase dan diulang sampai sistem yang lengkap siap.

#### 1. Fase Persyaratan

Pengembangan perangkat lunak adalah salah satu fase penting dan dalam pengembangan iterasi fase ini diulang secara berulang karena spesifikasi lengkap dari perangkat lunak dibagi menjadi potongan kebutuhan yang lebih kecil dan dikembangkan. Persyaratan iterasi ini dikumpulkan dan dipelajari dengan cermat untuk bekerja lebih lanjut dalam iterasi tertentu. Persyaratan ini bisa menjadi persyaratan baru atau perpanjangan dari persyaratan yang sudah dibangun

## 2. Fase Desain

Setelah kebutuhan terkumpul maka kita perlu mengimplementasikan tahap desain. Desain yang efektif diputuskan untuk mengimplementasikan persyaratan dari banyak alternatif. Ini adalah salah satu fase kritis karena desain yang tepat dapat memberikan hasil yang paling optimal akan mengurangi tekanan dana dari klien. Desain ini dapat berupa yang baru atau perluasan dari kebutuhan yang sudah dibangun.

## 3. Tahap Implementasi atau Pengembangan

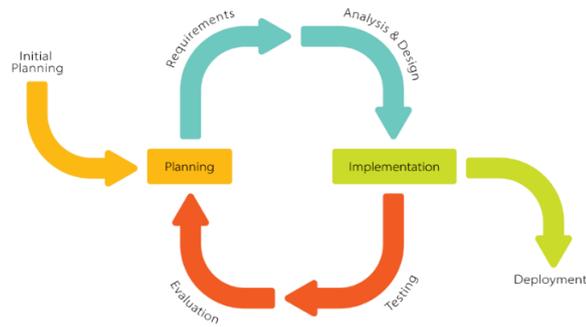
Desain yang telah diputuskan akan diimplementasikan oleh pengembang dengan standar coding dan metrik yang telah ditentukan. Pengembang perlu menerapkan pengujian unit pada setiap tahap pengembangan saat mengembangkan kode.

## 4. Pengujian

Setelah kode diimplementasikan maka fase pengujian ini diimplementasikan untuk mengidentifikasi setiap cacat yang ada dalam kode dan jika ada maka perlu dilaporkan kembali ke pengembang. Penguji dapat menulis kasus uji baru atau menggunakan yang sudah ada yang telah mereka tulis di build sebelumnya tetapi pengujian melalui adalah prioritas karena kesalahan apa pun akan berdampak pada spesifikasi perangkat lunak.

## 5. Fase Tinjauan

Pada fase ini, kebutuhan yang dikembangkan ditinjau untuk memenuhi semua standar sesuai dengan kebutuhan yang diputuskan saat ini. Berdasarkan rencana kebutuhan rencana lebih lanjut ini disusun dan diimplementasikan sebagai bagian dari siklus iterasi berikutnya. Berikut merupakan gambar dari tahapan SDLC model iteratif yang dapat dilihat pada gambar 2.3.



**Gambar 2.3 Tahapan model iteratif**

### 2.5.3 Model bentuk V

Model berbentuk V merupakan perpanjangan dari model air terjun. Model berbentuk V menunjukkan hubungan antara setiap fase pengembangan dan fase pengujian terkait. Aspek utama dari perangkat lunak apa pun adalah untuk melihat bagaimana kinerjanya. Itu perlu diuji berkali-kali. Oleh karena itu, fokus utama model berbentuk V adalah pada pengujian. Sementara verifikasi berfokus pada pengembangan, validasi memastikan bahwa semua pengembangan dilakukan dengan benar. Proses ini sangat dinamis dan banyak pengujian berlangsung. Model ini berguna ketika tidak ada persyaratan yang tidak diketahui, karena masih sulit untuk kembali dan melakukan perubahan. Untuk setiap fase yang sesuai, ia memiliki fase pengujian sehingga model terutama berfokus pada pengujian.

Model V berisi fase utama yang mirip dengan model proses lainnya, selain itu, model V juga memecah fase pengujian menjadi langkah-langkah terperinci untuk memastikan proses validasi dan verifikasi. fase pengujian dapat dilihat di bawah ini:

#### 1. Pengujian Unit

Pengujian unit adalah pengujian pada tingkat kode dan membantu menghilangkan masalah pada tahap awal, terutama pengembang bertanggung jawab untuk melakukan pengujian unit untuk kodenya sementara tidak semua cacat tidak dapat ditemukan pada pengujian unit.

## 2. Pengujian Fungsional

Pengujian fungsional dikaitkan dengan fase desain tingkat rendah yang memastikan bahwa kumpulan kode dan unit bekerja bersama mungkin untuk menjalankan fungsi atau layanan baru.

## 3. Tes integrasi

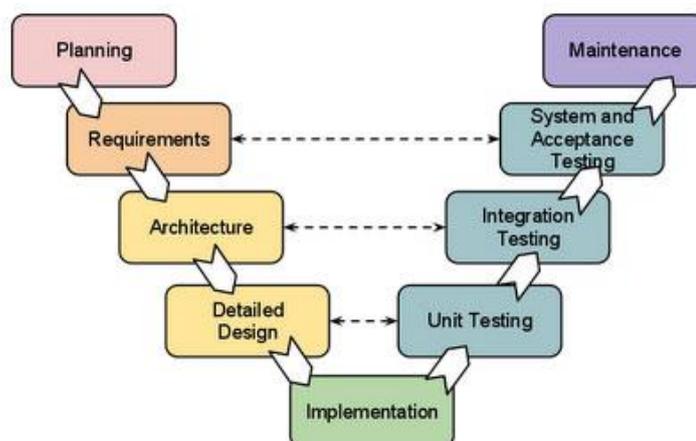
Pengujian integrasi dikaitkan dengan fase desain tingkat tinggi. Pengujian integrasi memastikan integrasi antara semua modul sistem setelah menambahkan fungsi atau pembaruan baru.

## 4. Pengujian Sistem

Pengujian sistem dikaitkan dengan persyaratan sistem dan fase desain. Ini menggabungkan perangkat lunak, perangkat keras, dan integrasi sistem ini dengan sistem eksternal lainnya.

## 5. Pengujian Penerimaan Pengguna

Pengujian Penerimaan Pengguna dikaitkan dengan fase analisis bisnis dan operasi. Pengguna pelanggan adalah pelaku utama pengujian ini berdasarkan kasus uji dan skenario yang mencakup persyaratan bisnis untuk memastikan bahwa mereka telah mengirimkan perangkat lunak yang tepat sesuai spesifikasi. Berikut merupakan gambar dari tahapan SDLC model V yang dapat dilihat pada gambar 2.4.



Gambar 2.4 Tahapan model bentuk V

#### **2.5.4 Model Agile**

Model *Agile* mengambil pendekatan iteratif untuk pengembangan perangkat lunak. Tidak seperti model air terjun linier langsung, proyek tangkas terdiri dari sejumlah siklus yang lebih kecil. Masing-masing dari mereka adalah proyek dalam bentuk mini: ia memiliki simpanan dan terdiri dari tahap desain, implementasi, pengujian, dan penerapan dalam lingkup pekerjaan yang telah ditentukan sebelumnya.. Jadi, tujuan utama dari model *Agile* adalah untuk memfasilitasi penyelesaian proyek dengan cepat. Untuk menyelesaikan perancangan, diperlukan kelincahan. Kelincahan dicapai dengan menyesuaikan proses dengan proyek, menghilangkan aktivitas yang mungkin tidak penting untuk proyek tertentu. Juga, segala sesuatu yang membuang-buang waktu dan usaha dihindari.

##### **1. Perencanaan konsep**

Pertama adalah fase konsep. Di sini, pemilik produk akan menentukan ruang lingkup proyek mereka. Jika ada banyak proyek, mereka akan memprioritaskan yang paling penting. Pemilik produk akan mendiskusikan persyaratan utama dengan klien dan menyiapkan dokumentasi untuk menguraikannya, termasuk fitur apa yang akan didukung dan hasil akhir yang diusulkan. Dianjurkan untuk menjaga persyaratan seminimal mungkin karena dapat ditambahkan pada tahap selanjutnya. Pada tahap konsep, pemilik produk juga akan memperkirakan waktu dan biaya proyek potensial. Analisis terperinci ini akan membantu mereka untuk memutuskan apakah suatu proyek layak atau tidak sebelum memulai pekerjaan.

##### **2. Analisis kebutuhan**

Setelah konsep diuraikan, sekarang saatnya untuk membangun tim pengembangan perangkat lunak. Pemilik produk akan memeriksa ketersediaan rekan kerja mereka dan memilih orang-orang terbaik untuk proyek tersebut sambil juga memberi mereka alat dan sumber daya yang diperlukan.

### 3. Desain

Selanjutnya adalah fase desain, juga disebut sebagai konstruksi. Ini cenderung menjadi fase terpanjang karena sebagian besar pekerjaan dilakukan di sini. Pengembang akan bekerja dengan desainer UX untuk menggabungkan semua persyaratan produk dan umpan balik pelanggan, mengubah desain menjadi kode.

### 2. Pengembangan

Produk hampir siap untuk dirilis. Tetapi pertama-tama, tim jaminan kualitas perlu melakukan beberapa tes untuk memastikan perangkat lunak berfungsi penuh. Anggota tim Agile ini akan menguji sistem untuk memastikan kodenya bersih, jika potensi bug atau cacat terdeteksi, pengembang akan mengatasinya dengan cepat.

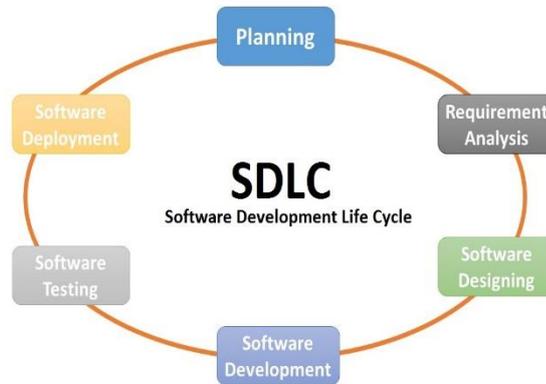
### 3. Pemeliharaan dan tes

Perangkat lunak sekarang akan sepenuhnya digunakan dan tersedia untuk pelanggan. Tindakan ini memindahkannya ke fase pemeliharaan. Selama fase ini, tim pengembangan perangkat lunak akan memberikan dukungan berkelanjutan untuk menjaga sistem berjalan dengan lancar dan menyelesaikan bug baru. Mereka juga akan siap menawarkan pelatihan tambahan kepada pengguna dan memastikan mereka tahu cara menggunakan produk. Seiring waktu, iterasi baru dapat dilakukan untuk menyegarkan produk yang ada dengan peningkatan dan fitur tambahan.

### 4. Pensiun

Ada dua alasan mengapa suatu produk akan memasuki fase pensiun: baik itu diganti dengan perangkat lunak baru, atau sistem itu sendiri menjadi usang atau tidak kompatibel dengan organisasi dari waktu ke waktu. Tim pengembangan perangkat lunak pertama-tama akan memberi tahu pengguna bahwa perangkat lunak akan dihentikan. Jika ada pengganti, pengguna akan dimigrasikan ke sistem baru. Terakhir, pengembang akan melakukan aktivitas akhir masa pakai yang tersisa dan menghapus dukungan untuk perangkat lunak yang ada. Berikut

merupakan gambar dari tahapan SDLC model *agile*. yang dapat dilihat pada gambar 2.5.



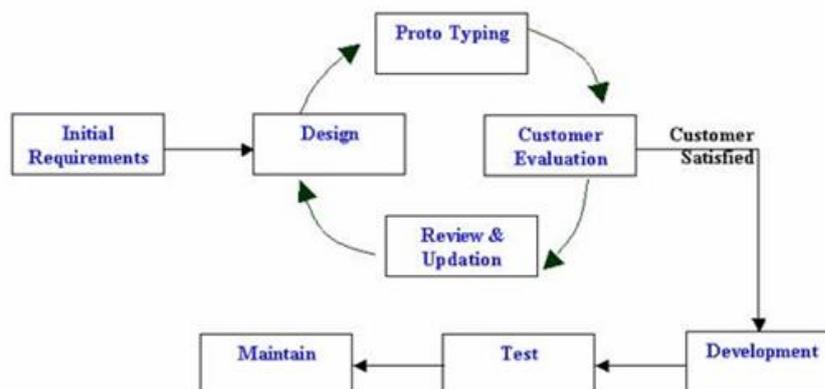
**Gambar 2.5 Tahapan model *agile***

### **2.5.5 Model *Prototype***

Prototipe atau model prototyping adalah model pengembangan perangkat lunak yang digunakan untuk membuat versi prototipe perangkat lunak. Model ini digunakan ketika pelanggan atau pengguna tidak memiliki informasi spesifik atau detail tentang produk. Dalam model ini, pengembang dapat mulai mengembangkan perangkat lunak dengan informasi dan persyaratan minimal kemudian mengambil umpan balik dari pengguna dan membuat perubahan sesuai dengan pengguna dan mendefinisikan kembali produk. Ini adalah proses berulang, coba-coba, dan kesalahan yang terjadi antara pengembang dan klien.

1. Persyaratan sistem baru atau harapan dari keluaran sistem diuraikan sedetail mungkin. Ini membutuhkan wawancara berbagai pengguna, yang mewakili semua segmen atau pemangku kepentingan dari sistem yang berlaku.
2. Sebuah spesifikasi tata letak awal dibentuk untuk sistem baru.
3. Model keluaran pertama dari sistem baru dibuat dari tata letak awal. Ini sering merupakan sistem yang diperkecil yang secara tentatif memberikan perkiraan output yang diinginkan yang diperlukan.

4. Pengguna memeriksa keluaran utama, mencatat kekuatan dan kelemahannya, hal-hal yang perlu dilakukan pada langkah selanjutnya dan hal-hal yang perlu dibuang. Pengembang mengumpulkan dan memeriksa komentar dari semua pemangku kepentingan.
5. Paradigma pertama diubah, didukung oleh komentar yang diberikan oleh pengguna, dan dibentuk menjadi keluaran kedua dari sistem baru.
6. Keluaran kedua dievaluasi dengan cara yang sama seperti keluaran utama.
7. Langkah-langkah ini diulang terus-menerus, sampai pengguna puas dengan hasilnya.
8. Sistem final karenanya dibangun, didukung oleh keluaran akhir. Sistem akhir sepenuhnya dievaluasi dan diuji. Pemeliharaan rutin dilakukan secara mulus untuk memperkirakan kegagalan skala besar dan untuk mengurangi periode waktu. Berikut merupakan gambar dari tahapan SDLC model *Prototype* yang dapat dilihat pada gambar 2.6.



Proto Type Model

Gambar 2.6 Tahapan model *prototype*

### **2.5.6 Model Spiral**

Model Spiral adalah salah satu model Siklus Hidup Pengembangan Perangkat Lunak yang paling penting, yang digunakan untuk manajemen risiko yang menggabungkan model air terjun dan model berulang. Model spiral pertama kali disebutkan oleh Barry Boehm dalam makalahnya tahun 1986. Dalam model ini, setiap fase dimulai dengan tujuan desain dan diakhiri dengan tinjauan kemajuan klien. Model ini digunakan untuk proyek-proyek besar yang melibatkan risiko dan biaya pada setiap perubahannya.

#### **1. Penetapan tujuan**

Setiap siklus dalam spiral dimulai dengan identifikasi tujuan siklus tersebut, berbagai alternatif yang mungkin untuk mencapai target, dan kendala yang ada.

#### **2. Penilaian dan Pengurangan Risiko**

Fase berikutnya dalam siklus ini adalah menghitung berbagai alternatif ini berdasarkan tujuan dan kendala. Fokus evaluasi pada tahap ini terletak pada persepsi risiko untuk proyek tersebut.

#### **3. Pengembangan dan validasi**

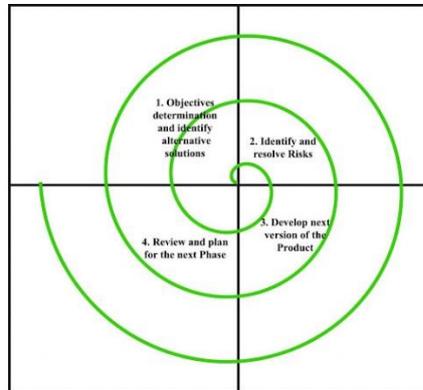
Fase berikutnya adalah mengembangkan strategi yang menyelesaikan ketidakpastian dan risiko. Proses ini dapat mencakup kegiatan seperti benchmarking, simulasi, dan prototyping.

#### **4. Perencanaan**

Langkah selanjutnya direncanakan. Proyek ini ditinjau, dan pilihan dibuat apakah akan melanjutkan periode spiral lebih lanjut. Jika bertekad untuk tetap, rencana disusun untuk langkah proyek selanjutnya.

Tahap pengembangan tergantung pada risiko yang tersisa. Misalnya, jika risiko kinerja atau antarmuka pengguna diperlakukan lebih penting daripada risiko pengembangan program, fase berikutnya mungkin merupakan pengembangan evolusioner yang mencakup pengembangan prototipe yang lebih rinci untuk

memecahkan risiko. Berikut merupakan gambar dari tahapan SDLC model spiral yang dapat dilihat pada gambar 2.7.



**Gambar 2.7 Tahapan model spiral**

## **2.6 Perencanaan sistem**

Tahap perencanaan sistem biasanya dimulai dengan permintaan formal ke departemen TI, yang disebut permintaan sistem, yang menggambarkan masalah atau perubahan yang diinginkan dalam sistem informasi atau proses bisnis. Di banyak perusahaan, perencanaan sistem TI merupakan bagian integral dari perencanaan bisnis secara keseluruhan. Ketika manajer dan pengguna mengembangkan rencana bisnis mereka, mereka biasanya menyertakan persyaratan TI yang menghasilkan permintaan sistem. Umumnya perencanaan sistem untuk memenuhi kepuasan pelanggan dengan menghasilkan produk dan jasa yang sesuai dengan kebutuhan konsumen, perencanaan sistem yang baik dapat tercapai jika perusahaan dapat mengintegrasikan seluruh produksi dan mendukung kegiatan dengan baik.[10].

Organisasi yang bergerak dibidang jasa, erat kaitannya dengan perubahan struktur pasar, produk dan khususnya teknologi, yang berpengaruh terhadap berkembangnya sebuah usaha. Oleh karena itu perencanaan yang baik juga didapat dari pengambilan keputusan yang juga baik, dan pemanfaatan sumberdaya sangat diperlukan, karena ini merupakan suatu pendekatan strategis terhadap peningkatan kinerja organisasi.[11].