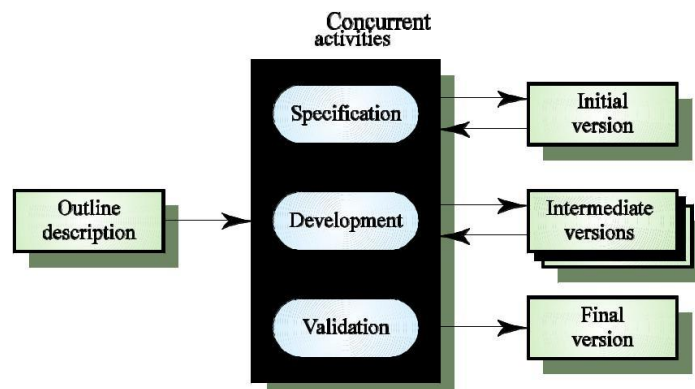


II.3 Teori Pendukung

Teori pendukung dibutuhkan untuk mendapatkan pemahaman yang baik mengenai sistem yang akan direalisasikan berkaitan dengan perangkat lunak dan teknologi-teknologi pendukung yang dapat direalisasikan.

II.3.1 Pengembangan Sistem Evolusioner

Metode pengembangan evolusioner berdasarkan pada ide untuk mengembangkan implementasi awal, kemudian memperlihatkan sistem awal itu kepada *user* untuk dikomentari, dan memperbaikinya versi demi versi sampai sistem yang memenuhi persyaratan diperoleh. Pada metode ini tidak ada kegiatan spesifikasi, pengembangan, dan validasi yang terpisah. Kegiatan-kegiatan ini dilakukan pada saat yang bersamaan dengan umpan balik yang cepat untuk masing-masing kegiatan.



Gambar II.3.1 : Model pengembangan system evolusioner

Pengembangan system secara evolusioner sendiri memiliki 2 variasi : **Pengembangan Eksplotari**, Tujuan proses ini adalah bekerja dengan pelanggan untuk menyelidiki persyaratan mereka dan mengirimkan sistem akhir. Harusnya diawali dengan kebutuhan yang sudah dimengerti. Pendekatan ini biasa digunakan untuk mengembangkan system berdasar pesanan dari klien (custom product). Variasi kedua adalah **Prototipe yang dapat dibuang** (*throw-away-prototype*), metode ini berkonsentrasi pada eksperimen, dengan persyaratan pelanggan yang tidak dipahami dengan baik, cocok untuk pengembangan software jenis generic product.

Ada beberapa kelebihan metode evolusioner jika dibandingkan dengan SDLC. Metode evolusioner lebih efektif dari pendekatan SDLC dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan. Karena untuk menghasilkan sistem informasi tidak harus menempuh keseluruhan tahap SDLC. Keuntungan kedua, sementara user mendapat pemahaman yang lebih baik dari masalah mereka, sistem perangkat lunak dapat merefleksikannya.

Adapun kelemahan dari sistem ini antara lain :Kurangnya visibilitas proses,jika sistem dikembangkan dengan cepat, tidaklah efektif dari segi biaya jika dihasilkan dokumen yang merefleksikan setiap versi sistem. Kedua, sistem seringkali memiliki struktur yang buruk hal ini disebabkan perubahan yang terus-menerus pada program cenderung merusak struktur perangkat lunak. Hal ini berakibat pada penyesuaian perubahan menjadi kian sulit dan mahal. Ketiga metode ini membutuhkan kemampuan pengembangan software dengan SDM yang sudah berpengalaman [6].

II.3.2 Hubungan Rancangan Database

Perancangan Database adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem.

Perancangan sistem terjadi pada dua tingkat , yaitu :

Pada tingkat pertama, perencanaan sistem, analisis dan rancangan umum dilaksanakan untuk menetapkan kebutuhan pemakai. Tingkat perancangan database ini melibatkan tahap *front-end*, bebas dari perancangan database tertentu atau Database Management System (DBMS).

Pada tingkat kedua, rancangan umum, seperti diagram entitas relasi tingkat tinggi, ditransformasikan (atau didekomposisikan) ke dalam perancangan database rinci untuk sebuah DBMS tertentu yang akan digunakan untuk mengimplementasikan sistem total.

Pada masa lalu banyak penjual (*vendors*) menawarkan Database Management Systems (DBMS) yang berdasarkan pada Model Hierarkikal dan Model Jaringan. Saat ini Model Relasional adalah dominan. Karena itu hampir semua penjual perangkat lunak database menawarkan produk perangkat lunak Relational Database Management Systems (RDBMS).

RDBMS dibuat dengan struktur tiga skema, Struktur lapisan ini mendefinisikan data perusahaan pada tingkat yang berbeda.

Skema Eksternal mendefinisikan bagaimana pemakai mengakses dan melihat output dari RDBMS, bebas dari bagaimana data disimpan atau diakses secara fisik. Akses dan manipulasi seperti ini dilaksanakan oleh pemakai dengan memperkerjakan bahasa prosedural, seperti COBOL atau bahasa query, seperti Structured Query Language (SQL), bahasa standar yang diakui untuk RDBM [7]

