BAB II

LANDASAN TEORI

Pada bab ini dijelaskan beberapa konsep dan dasar teori yang berkaitan dengan permasalahan yang akan dibahas penyusun sebagai dasar pemahaman dalam mengimplementasikan konsep-konsep tersebut ke dalam semua kegiatan pengembangan sistem yang akan dibuat.

2.1. Konsep Dasar Sistem

Menurut Abdul Kadir (2003 : 54) pada dasarnya, sistem adalah sekumpulan elemen yang saling berkaitan atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.

Sedangkan menurut Al- Bahra Bin Ladjamudin (2005 : 3) sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan dan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Berdasarkan definisi diatas maka sistem adalah kumpulan suatu jaringan yang saling berinteraksi atau saling berhubungan untuk melakukan suatu kegiatan untuk mencapai suatu tujuan yang diharapkan.

2.1.1. Karakteristik Sistem

Menurut Al-Bahra Bin Ladjamudin (2005 : 3) Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, antara lain :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar sistem dari suatu sistem adalah apaun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energy dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengangu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainya. Melalui penghubung ini kemungkinan

sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainya membentuk satu kesatuan.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance input* adalah energy yang dimasukan supaya sistem tersebut dapat berjalan. Sinyal *input* adalah energy yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

Keluaran sistem adalah energi yang di olah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak aka nada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.2. Definisi Informasi

Menurut Al-Bahra Bin Ladjamudin (2005 : 8) informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang.

Sedangkan menurut Zulkifli Amsyah, MLS (2005 : 289), informasi adalah bahan yang dihasilkan dari pengolahan data. Data berorientasi pada kegiatan operasional, seperti transaksi misalnya. Informasi berorientasi pada kegiatan manajemen, baik lini bawah, lini tengah, maupaun lini atas.

Jadi informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan.

Kegunaan informasi adalah untuk mengurangi ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Inforamsi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Informasi digunakan tidak hanyaoleh satu pihak didalam organisasi. Nilai sebuah informasi ditentukan dari dua hal yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut.

2.2.1. Konsep Dasar Informasi

Merupakan konsep yang mendasari timbulnya informasi, maka penjabaran untuk mengetahui apa itu informasi dan elemen yang terkandungnya adalah sebagai berikut :

2.2.1.1. Data

Menurut Al-Bahra Bin Ladjamudin (2005 : 8) data adalah kumpulan saru suatu kenyataan yang berupa angka, huruf, suara, symbol, atau gabungan yang belum memiliki arti atau belum dapat bercerita.

2.2.1.2. Pengolahan Data (*Data Processing*)

Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan. Ada beberapa operasi yang dilakukan dalam pengolahan data, antara lain :

1. Data Masukan

Kumpulan data transaksi kesebuah pengolahan data medium, merupakan data masukan. Contoh lain dari data masukan adalah pengkodean data transaksi kedalam bentuk lain (contoh, *converting* atribut kelamin *female* ke huruf F), dan penyortiran data atau informasi untuk pengambilan keputusan *(potential information for future)*.

2. Data Transformasi

Beberapa bentuk data transformasi diantaranya adalah sebagai berikut:

a. Kalkulasi operasi aritmatik terhadap data field.
b. Menyimpulkan proses akumulasi beberapa data, misalkan, menjumlah jumlah jam kerja setiap hari dalam seminggu menjadi nilai total jam kerja per minggu.

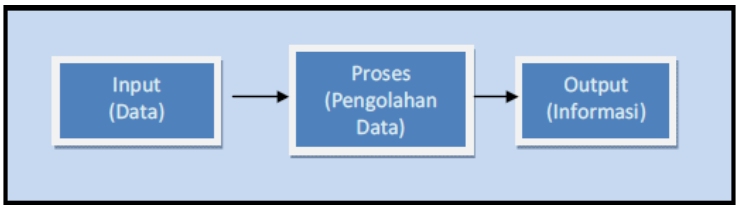
3. Informasi Keluaran

Menampilkan hasil merupakan kegiatan untuk menampilkan informasi yang dibutuhkan pemakai melalui monitor atau cetakan, sedangkan *reproducing* (memproduksi ulang) merupakan kegiatan penyimpanan data yang digunakan untuk pemakai lain yang membutuhkan. *Telecommunicating* (Telekomunikasi) adalah kegiatan penyimpanan data secara elektronik melalui saluran komunikasi.

2.2.2. Siklus Informasi

Menurut Al-Bahra Bin Ladjamudin (2005 : 11) untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi.

Siklus informasi atau siklus pengolahan data adalah sebagai berikut :



Gambar 2.1 Komponen Sistem Informasi
(Sumber : **Al-Bahra Bin Ladjamudin.** 2005. *Analisis dan Desain Sistem Informasi*. Graha Ilmu. Yogyakarta)

2.2.3. Nilai Informasi

Menurut Zulkifli Amsyah (2005 : 316) nilai informasi ditentukan oleh lima karakteristik, yaitu :

1. Ketelitian (*Accuracy*)

Ketelitian atau akurasi dapat didefinisikan sebagai perbandingan dari informasi yang benar dengan jumlah seluruh informasi yang dihasilkan pada satu proses pengolahan data tertentu.

2. Ketepatan Waktu (*Timeliness*)

Ketepatan waktu merupakan karakteristik informasi lainnya yang penting. Bukan hanya bernilai baru atau lama, tetapi tepat waktu atau setidaknya saat informasi diperlukan. Knedasti informasinya akurat tetapi jika diterimanya atau diketahuinya terlambat tentu saja sudah tidak berguna lagi.

3. Kelengkapan (*Complete*)

Kadang-kadang kita menghadapi suatu keputusan yang harus dibuat dengan informasi yang teliti, waktu yang tepat, tetapi informasi tidak lengkap. Sering kali kegiatan bisnis yang memerlukan pengambilan keputusan secara cepat menjadi tertunda hanya karena kurang lengkapnya informasi yang ada.

4. Ringkas (*Conciseness*)

Karena sering menghadapi masalah kurang lengkapnya informasi, maka sering terjadi dalam penyediaan suatu informasi diupayakan secara berlebihan. Padahal informasi yang bernilai lebih cenderung bersifat seperti suatu kesimpulan, dan akan lebih jelas dan bernilai tinggi bila dapat disertai dengan bagan, gambar, tabel, dan bentuk statistik lainnya.

5. Kesesuaian (*Relevancy*)

Informasi yang bernilai tinggi tentu saja mempersyaratkan pula unsur kelima yaitu sesuai (relevan). Informasi hedaklah sesuai dengan keperluan

pekerjaan atau keperluan manajemen. Data yang sama sering kali perlu diolah secara berbeda untuk memperoleh informasi yang sesuai dengan keperluan unit masing-masing.

2.3. Definisi Pelayanan

Menurut Kamus Besar Bahasa Indonesia (2005 : 646) pelayanan adalah 1 perihal atau cara melayani; 2 usaha melayani kebutuhan orang lain dengan memperoleh imbalan (uang); jasa; 3 kemudahan yang diberikan sehubungan dengan jual beli barang atau jasa.

Produksinya dapat dikaitkan atau tidak dikaitkan pada satu produk fisik sehingga pelayanan merupakan perilaku produsen dalam rangka memenuhi kebutuhan dan keinginan konsumen demi tercapainya kepuadan pada konsumen sendiri.

Jadi dapat didefinisikan pelayanan merupakan suatu kegiatan yang diberikan seseorang atau badan untuk melayani kebutuhan orang lain. Apabila pelayanan yang ditawarkan perusahaan sesuai dengan keinginan konsumen maka produk/jasa yang ditawarkan akan dibeli. Sedangkan bila terjadi pelayanan yang tidak sesuai dengan keinginan konsumen maka dapat di pastikan produk/jasa tersebut kurang diminati konsumen.

2.3.1. Kualitas Pelayanan

Bagi perusahaan yang bergerak di bidang jasa, kualitas pelayanan merupakan faktor yang sangat penting. Karena dalam memasarkan produk jasa, interaksi antara produsen dan konsumen terjadi secara langsung. Aplikasi kualitas pelayanan sebagai sifat dari penampilan produk atau kinerja merupakan salah satu bagian utama dari strategi perusahaan dalam meraih keunggulan yang

berkesinambungan. Baik sebagai pemimpin pasar atau sebagai strategi untuk terus berkembang.

Menurut Goetsh dan Davis (Fandy Tjiptono, 2000: 81) bahwa kualitas pelayanan adalah merupakan kondisi dinamis yang berhubungan dengan produk jasa, dan manusia proses dan lingkungan yang memenuhi atau melebihi pelanggan.

Sedangkan menurut Hary (Fandy Tjiptono, 2000: 90) kualitas pelayanan merupakan suatu proses atau aktivitas yang dilakukan oleh perusahaan yang dapat dirasakan secara langsung hasilnya, yang pada akhirnya memenuhi harapan pelanggan.

Pelayanan yang berkualitas dan memenuhi kepuasan konsumen terdiri dari tiga komponen dasar yang harus dipahami bagi setiap perusahaan yaitu:

a. Proses Sebelum Penjualan

Pada tahap ini perusahaan mempunyai kesempatan untuk membentuk hubungan dengan konsumen. Hal ini dapat dilakukan dengan menginformasikan produk pada konsumen dan menciptakan kepercayaan pada konsumen atas produk yang ditawarkan.

b. Proses Selama Transaksi

Pada fase ini perusahaan harus tetap menjaga kualitas pelayanan. Agar konsumen tetap menjadi pelanggan setia. Komunikasi pada fase ini sangat penting. Sebab pada fase ini konsumen membutuhkan informasi lebih banyak lagi akan produk yang dibeli. Jika pihak pihak perusahaan tidak memahami tentang produk yang ditawarkan bisa saja konsumen beralih pada perusahaan lain.

c. Proses Sesudah Penjualan

Pada fase ini perusahaan diharapkan mendengar atau menanggapi keluhan dari pihak konsumen atas produk yang telah dibeli.

2.4. Definisi Persediaan Barang

Mennurut Bambang Riyanto (2000 : 122) menyatakan bahwa persediaan adalah barang-barang yang dimiliki oleh perusahaan pada suatu saat tertentu dengan maksud dijual kembali baik secara langsung maupun melalui proses produksi dalam sirkulasi operasi normal perusahaan dalam hal ini termasuk pula barang-barang yang masih dalam proses produksi atau menunggu untuk digunakan.

2.4.1. Sistem Persediaan Barang

Sistem persediaan barang adalah suatu struktur interaksi antara manusia, peralatan, metode-metode, dan pengendalian yang disusun untuk mencapai tujuan-tujuan seperti berikut ini :

1. Mendukung pekerjaan rutin dalam berbagai pengendalian persediaan barang.
2. Mendukung pembuatan keputusan untuk karyawan yang mengelola gudang dan bagian penyimpanan.
3. Mendukung persiapan persediaan barang.

Sistem persediaan barang berguna untuk mencatat data yang berhubungan dengan persediaan barang, misalnya transaksi penerimaan dan pengeluaran barang, transaksi penjualan, transaksi penyesuaian persediaan barang yang terdiri dari pengembalian, penjualan, penurunan, dan penaikan harga persediaan barang.

2.4.2. Jenis-jenis Persediaan

Menurut Isroah (2005:48) jenis persediaan dapat diklasifikasikan menjadi beberapa bagian yaitu :

1. Bahan Baku (*Raw Materials*)

   Merupakan bahan yang diperoleh untuk digunakan dalam proses. Bahan baku ini dapat diperoleh langsung dari sumber alam ataupun melalui pembelian dari perusahaan lain yang merupakan barang jadi.

2. Barang Dalam Proses (*Work In Process*)

   Merupakan barang yang masih memerlukan proses lebih lanjut sebelum menjadi barang jadi yang dapat dijual.

3. Barang Jadi (*Finished Goods*)

   Merupakan barang yang sudah selesai dikerjakan dalam proses produksi dan siap untuk dijual.

4. Bahan Pembantu (*Supplies*)

   Bahan ini meliputi :

   a. Bahan yang bukan merupakan bagian atau komponen dari barang jadi, tetapi diperlukan untuk membantu berhasilnya suatu proses produksi.
   b. Bahan yang merupakan bagian dari barang jadi, tetapi tidak ekonomis untuk ditelusuri pemakaiannya pada setiap jenis proses produksi.

2.5. Definisi *Waiting List*

Waiting List atau daftar tunggu adalah orang yang sedang menunggu giliran pelayanan terdaftar pada daftar tunggu untuk mendapatkan pelayanan

selanjutnya ketika dipanggil (misal untuk memperoleh sambungan telepon, mendapat tempat di pesawat, mendapat perawatan kesehatan).

2.6. Metode Penelitian

Metodel penelitian menurut Sugiyono (2011 : 3), pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Data yang diperoleh melalui penelitian itu adalah data empiris (teramati) yang mempunyai kriteria tertentu yaitu valid, reliable, dan obyektif.

Metode penelitian ini merupakan bagian dari metodologi yang secara khusus mendeskripsikan tentang cara mengumpulkan dan menganalisis data. Setiap penelitian mempunyai tujuan dan kegunaan tertentu. Secara umum tujuan penelitian ada tiga macam yaitu yang bersifat penemuan, pembuktian, dan pengembangan. Dalam upaya mencapai tujuan dari sebuah penelitian diperlukan metode yang sesuai dan mudah.

Menurut Sugiyono (2011 : 29), bahwa berdasarkan tingkat eksplanasinya terdapat 3 macam metode penelitian yaitu sebagai berikut :

a. Penelitian Deskriptif

Penelitian deskriptif adalah suatu metode yang digunakan untuk menggambarkan atau menganalisis suatu hasil penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas.

b. Penelitian Komparatif

Penelitian komparatif adalah suatu penelitian yang bersifat membandingkan. Disini variabelnya masih sama dengan variabel mandiri tetapi untuk sample yang lebih dari satu, atau dalam waktu yang berbeda.

c. Penelitian Asosiatif

Penelitian asosiatif merupakan penelitian yang bertujuan untuk mengetahui pengaruh ataupun juga hubungan antara dua variabel atau lebih. Penelitian ini mempunyai tingkatan tertinggi dibandingkan dengan diskriptif dan komparatif karena dengan penelitian ini dapat dibangun suatu teori yang dapat berfungsi unguk menjelaskan, meramalkan dan mengontrol suatu gejala.

2.7. Desain Penelitian

Menurut Moh. Nazir (2003 : 84) memaparkan bahwa desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian.

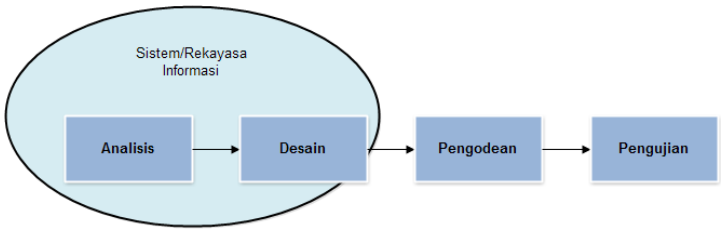
Keseluruhan dari perencanaan bertujuan untuk menjawab pertanyaan penelitian dan mengantisipasi beberapa kesulitan yang mungkin timbul selama proses penelitian, hal ini penting karena desain penelitian merupakan strategi untuk mendapatkan data yang dibutuhkan untuk keperluan pengujian hipotesis atau untuk menjawab pertanyaan penelitian dan sebagai alat untuk mengontrol variabel yang berpengaruh dalam penelitian.

2.8. Model-Model Pengembangan Sistem

Terdapat beberapa metode dalam melakukan pengembangan sistem, namun dalam hal ini hanya akan membahas dua metode yang mendekati dalam melakukan pengembangan sistem. Adapun dua metode yang mendekati yaitu sebagai berikut :

2.8.1. Model *Waterfall*

Model *waterfall* sering juga disebut sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Gambar 2.2 Model *Waterfall*
(Sumber : **Rosa A.S – M.Shalahuddin.** 2011. *Rekayasa Perangkat Lunak (Terstuktur dan Berorientasi Objek)*. Modula. Bandung)

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap

selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga peril didokumentasikan.

c. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian focus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

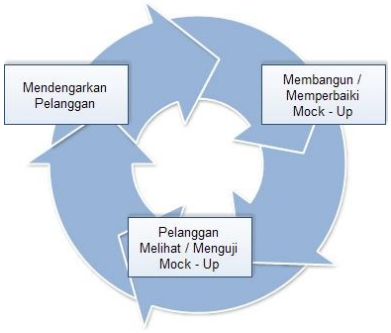
Dari kenyataannya yang terjadi sangat jarang model *waterfall* dapat dilakukan sesuai alurnya karena :

a. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur perkembangan.

Dengan berbagai kelemahan yang dimiliki model *waterfall* tapi model ini telah menjadi dasar dari model-model lain yang dalam melakukan perbaikan model pengembangan perangkat lunak.

2.8.2. Model Prototype

Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.



Gambar 2.3 Model *Prototype*
(Sumber : **Rosa A.S – M.Shalahuddin.** 2011. *Rekayasa Perangkat Lunak (Terstuktur dan Berorientasi Objek)*. Modula. Bandung)

Model prototipe dimulai dari mengumpulkan kebutuhan pelanggan tehadap perangkat lunak yang akan dibuat. Lalu dibuatlan program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user.

Model prototipe juga memiliki kelemahan sebagai berikut :

a. Pelanggan dapat sering mengubah-ubah ata menambah sepesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan

pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.

b. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembangan lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iterasi tanpa akhir.

Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek dapat berjalan sesuai dengan target waktu dan biaya di awal, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek.

2.9. Metode Pendekatan

Pendekatan sistem dapat diartikan sebagai titik tolak atau sudut pandang kita terhadap proses sebuah sistem yang merujuk pada pandangan tentang terjadinya suatu proses yang bersifat masih sangat umum dalam mewadahi, mengispirasi, dan melatari metode pendekatan dalam cakupan teoritis tertentu.

Didalam melakukan penelitian ini, penulis menggunakan metode pendekatan sistem berorientasi objek, namun ada baiknya apabila kita melakukan perbandingan terlebih dahulu antara pendekatan terstruktur dan pendekatan berorientasi objek.

2.9.1. Pendekatan Terstruktur

Pada pendekatan terstruktur ini permasalaha dilihat sebagai urutan sesuatu yang harus dikerjakan, seperti menerima masukan, pemrosesan, kemudian menghasilkan keluaran. Apa yang dilakukan pada pendekatan terstuktur terutama adalah menulis daftar perintah yang harus diikuti oleh komputer, kemudian mengorganisasikan perintah-perintah tersebut kedalam kelompok-kelompok yang dinamakan fungsi / prosedur.

Kelemahan pendekatan terstuktur adalah data global, pada program yang berukuran sangat besar sangat sulit untuk dilacak. Selain itu kelemahannya yaitu tidak menggambarkan dunia nyata (*real world*) dengan baik. Sebabnya adalah karena fungsi-fungsi berorientasi pada aksi dan tidak terhubung langsung dengan permasalahan. Pendekatan ini lebih memungkinkan perangkat lunak sesuai dengan keinginan pengembang, bukan sesuai dengan kebutuhan/keinginan pengguna.

Sifat- sifat dari pemrograman terstruktur adalah sebagai berikut :

1. Memuat teknik pemecahan masalah yang logis dan sistematis.
2. Memuat algoritma yang efesien, efektif dan sederhana.
3. Program disusun dengan logika yang mudah dipahami.

4. Tidak menggunakan perintah GOTO.
5. Biaya pengujian program relatif rendah.
6. Memiliki dokumentasi yang baik.
7. Biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah

2.9.2. Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perngkat lunak, sistem informasi, atau sistem lainnya). Pendekatan ini akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya dan dapat berinteraksi satu sama lain.

Metode berorientasi objek dapat mengembangkan aplikasi yang saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility*, dan sebagainya) dengan *platform* yang berbeda-beda,sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Menurut Adi Nugroho (2005 : 30), alat-alat yang digunakan dalam pendekatan analisis dan pemrograman berorientasi objek yaitu dengan notasi UML dengan membuat tujuh diagram yaitu:

a. *Use case diagram*

   Diagram ini bersifat statis dimana diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

b. *Activity diagram*

   Diagram ini bersifat dinamis dengan memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan member tekanan pada aliran kendali antar objek.

c. *Sequence diagram*

   Diagram ini bersifat dinamis, dalam interaksinya menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

d. *Collaboration diagram*

   Diagram ini bersifat dinamis, dalam interaksinya menekankan pada organisasi structural dari objek-objek yang menerima serta mengirim pesan (*message*).

e. *Class diagram*

   Diagram ini bersifat statis, yang memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi, serta relasi. Diagram ini umum dijumpai pada

pemodelan sistem berorientasi objek. Meskipun bersifat statis sering pula diagram kelas ini memuat kelas-kelas aktif.

f. *Component diagram*

Diagram ini bersifat statis, yang memperlihatkan organsisasi serta ketergantungan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas, antarmuka, serta kolaborasi.

g. *Deployment diagram*.

Diagram ini bersifar statis, yang memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run-time*). Diagram ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada didalamnya. *Deployment* diagram sangat erat hubungannya dengan diagram komponen dimana *deployment* diagram memuat satu arau lebih komponen.

2.10. Bahasa Pemrograman *Java*

Java adalah bahasa pemrograman yang berorientasi objek (OOP). *Java* diciptakan oleh *James Gosling*, *developer* dari *Sun Microsystems* pada tahun 1991. Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. Sebagai bahasa pemrograman yang berorientasi objek, *Java* menggunakan kelas untuk membentuk suatu objek. Bahasa pemrograman *Java* pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak.

Java adalah bahasa pemrograman yang multi *platform* dan *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang *support Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam *p-code* dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan *platform* sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

2.10.1. Keunggulan *Java*

Java mempunyai beberapa keunggulan dibandingkan dengan bahasa pemrograman lainnya. Keunggulan bahasa pemrograman Java antara lain:

a. Berorientasi Objek

Java adalah bahasa pemrograman yang berorientasi pada objek. *Java* membagi program menjadi objek-objek serta memodelkan sifat dan tingkah laku masing-masing dalam menyelesaikan suatu masalah.

b. *Java* bersifat *Multiplatform*

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan berbeda. Untuk mengakomodasi hal tersebut, *Java compiler* membangkitkan *bytecodes* (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak *platform* perangkat keras dan perangkat lunak secara efisien). *Java* dapat dijalankan oleh banyak platform seperti Linux, Unix, Windows, Solari, maupun Mac.

c. *Java* bersifat *Multithread*

Multithreading adalah kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. *Thread* dalam *Java* memiliki kemampuan untuk memanfaatkan kelebihan multi *prosessor* apabila sistem operasi yang digunakan mendukung multi prosessor.

d. Dapat didistribusi dengan mudah

Java memiliki library rutin yang lengkap untuk dirangkai pada *protocol* TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan *networking Java* lebih kuat dan lebih mudah digunakan. *Java* memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah soket koneksi. *Java* juga mamudahkan pembuatan CGI (*Common Gateway Interface*).

e. Bersifat Dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang. *Java* bersifat dinamis dalam tahap linking. *Class* yang ada dapat di link sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di link dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

2.10.2. Kekurangan *Java*

Kekurangan yang dimiliki oleh *Java* adalah pada satu slogannya, takni "Tulis sekali dan jalankan dimana saja" ternyata tidak sepenuhnya benar. Beberapa hal harus disesuaikan jika dijalankan pada *platform* yang berbeda. Misalnya untuk J2SE dengan *platform* SWT-AWT *bridge* tidak dapat berfungsi di Mac OS X. Kekurangan lainnya adalah kemudahan aplikasi *Java* didekompilasi.

Dekompilasi adalah suatu proses membalikkan sebuah aplikasi menjadi kode sumbernya. Hal ini memungkinkan terjadi pada *Java* karena berupa *bytecode* yang menyimpan bahasa tingkat tinggi. Hal ini terjadi pula pada *platform*. NET dari Microsoft sehingga program yang dihasilkan mudah dibajak kodenya karena sulit untuk disembunyikan.

2.11. Perangkat Lunak Pendukung

Perangkat Lunak (*Software)* pendukung sangatlah dibutuhkan dalam membangun sistem aplikasi ini, karena sistem aplikasi yang akan dibangun ini membutuhkan beberapa program aplikasi yang digunakan untuk menghasilkan program aplikasi yang lengkap yang sesuai dengan yang dibutuhkan oleh pengguna. Adapun program aplikasi yang digunakan oleh program aplikasi yang akan dibangun ini adalah *Netbeans* dan My*SQL*.

2.11.1. *NetBeans*

NetBeans dimulai pada tahun 1996 sebagai Xelfi (kata bermain pada Delphi), *Java* IDE proyek mahasiswa di bawah bimbingan Fakultas Matematika dan Fisika di *Charles University* di Praha. Pada tahun 1997 Staněk Romawi membentuk perusahaan sekitar proyek tersebut dan menghasilkan versi komersial *NetBeans* IDE hingga kemudian dibeli oleh *Sun Microsystems* pada tahun 1999. Komunitas *NetBeans* sejak terus tumbuh, berkat individu dan perusahaan yang menggunakan dan berkontribusi dalam proyek ini.

NetBeans mengacu pada kedua platform kerangka untuk aplikasi desktop *Java*, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan

dengan *Java, JavaScript, PHP, Python, Ruby, Groovy,* C, C++, *Scala, Clojure,* dan lain-lain.

NetBeans IDE ditulis dalam *Java* dan berjalan di mana-mana mana JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas *Java*, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.

Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen *software* yang disebut modul. Aplikasi berbasis *platform NetBeans* (termasuk IDE *NetBeans*) dapat diperpanjang oleh pengembang pihak ketiga.

Fitur fitur yang terdapat dalam netbeans antara lain:

1. *Smart Code Completion*: untuk mengusulkan nama variabel dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
2. *Bookmarking*: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. *Go to commands*: fitur yang digunakan untuk *jump* ke deklarasi variabel, *source code* atau file yang ada pada *project* yang sama.
4. *Code generator*: jika kita menggunakan fitur ini kita dapat meng-*generate constructor, setter* and *getter method* dan yang lainnya.
5. *Error stripe*: fitur yang akan menandai baris yang *error* dengan memberi *highlight* merah.

2.11.2. *MySQL*

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan *database* bernama *MYSQL* AB yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama TcX Data Konsult AB, dan tujuan awal dikembangkannya *MySQL* adalah untuk mengembangkan aplikasi berbasis *web* pada client. Awalnya Michael "Monty" Widenius, pengembang satu-satunya di TcX memiliki sebuah aplikasi UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka *SQL* yang cocok untuk diimplementasikan ke dalamnya.

Mula-mula *Monty* memakai *miniSQL* (*mSQL*) pada eksperimennya itu, namun SQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan *query*. Akhirnya *Monty* menghubungi David Hughes, pembuat *mSQL* yang sedang merilis versi kedua dari *mSQL*. Kemudian *Monty* mencoba membuat sendiri mesin *SQL* yang memiliki antarmuka mirip dengan *SQL*, tetapi dengan kemampuan yang lebih sesuai sehingga lahirlah *MySQL*. Tentang pengambilan nama *MySQL*, sampai saat ini masih belum jelas asal usulnya. Ada yang berpendapat nama *My* diambil dari huruf depan dan belakang *Monty*, tetapi versi lain mengatakan nama itu diambil dari putri *Monty* yang kebetulan juga bernama *My*.

2.11.3. *IReport*

Tool untuk reporting yang paling populer di *Java* adalah *iReport*. *iReport* penggunaanya mirip dengan *Crystal Report* yang sudah banyak dikenal orang.

Dengan *iReport* kita dapat membuat *report* dengan mudah. Cukup dengan drag dan *drop* dan juga *wizard* yang sangat mudah digunakan. *iReport* sendiri tersedia dalam bentuk aplikasi tunggal dan yang paling baru adalah *iReport* sudah dapat diintegrasikan dengan *Netbeans* IDE. Artinya kita tidak perlu lagi menggunakan 2 *tools* yang berbeda untuk membuat aplikasi yang lengkap dengan reportnya.