

BAB II

LANDASAN TEORI

2.1 Sistem

(Effendi, 2020) menyatakan bahwa sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Secara garis besar ada dua kelompok pendekatan sistem, yaitu Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya didefinisikan sebagai Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu.

2.2 Metode Pengumpulan Data

Dalam penelitian ini, peneliti menggunakan 3 teknik pengumpulan data, yaitu :

1. Wawancara

Wawancara merupakan salah satu usaha atau percobaan secara sistematis, untuk mengumpulkan informasi yang kita butuhkan, yaitu dengan cara memberikan pertanyaan-pertanyaan kepada pihak yang menurut kita dapat dijadikan sumber data yang relevan dengan penelitian kita. Tujuan dari wawancara adalah untuk memperoleh informasi yang lebih akurat dan lengkap, untuk menyusun sistem yang baru agar sesuai dengan kebutuhan sistem organisasi.

2. Observasi

Observasi merupakan pengumpulan data yang dilakukan dengan cara mendatangi langsung dan mempelajari sistem yang terjadi secara nyata ditempat kejadian.

3. Studi Pustaka

Studi pustaka yaitu salah satu teknik pengumpulan data yang dilakukan dengan cara membaca, mengutip, dan membuat serta mempelajari catatan, buku-buku dan literatur-literatur yang bersumber pada bahan-bahan pustaka yang mendukung dan dapat dijadikan bahan referensi terkait dengan penelitian.

2.3 Analisis PIECES

Pengertian analisis pieces adalah suatu system yang di gunakan untuk analisis system kerja pada suatu perusahaan atau organisasi. Ada 6 kriteria analisis pieces yaitu kinerja

(*Performance*), informasi (*Information*), ekonomi (*Economic*), kontrol (*Control*), efisiensi (*Efficiency*), dan pelayanan (*Services*).

Analisis pieces juga sangat banyak diterapkan untuk penelitian pada suatu perusahaan atau organisasi. Selain mudah dan dapat dipahami analisis pieces juga bersifat ringan tidak membutuhkan data yang banyak.

Berikut ini kriteria yang wajib ada pada analisis pieces menurut (Wetherbe, 2018) :

1. *Performance*

- Produksi

Jumlah kerja selama periode waktu tertentu. Pada bagian ini dideskripsikan situasi saat ini tentang jumlah kerja yang dibutuhkan untuk melakukan serangkaian kerja tertentu dalam satuan orang jam, orang hari, atau orang bulan. Misalnya : untuk memperioses berkas yang masuk kepada oraganisasi dibutuhkan berapa orang jam? Kemudian hal ini dianalisis apakah hasil kerja yang demikian ini sudah bagus atau perlu ada peningkatan kerja.

- Waktu *respons*

Penundaan rata-rata antara transaksi atau permintaan dengan *respons* ketransaksi atau permintaan tersebut. Pada bagian ini dideskripsikan situai saat ini tentang waktu *respons* yang terjadi ketika ada suatu transaksi yang masuk hingga transaksi tersebut *direspons* untuk diproses. Penundaan ini bisa jadi karena antrian dalam pemrosesan transaksi-transaksi sebelumnya

2. *Informations*

Pada bagian ini dideskripsikan pada situasi saat tentang kurangnya informasi yang dibutuhkan untuk pengambilan keputusan, baik itu dalam jumlah, informasi maupun dalam hal macam informasinya.

Terlalu banyak informasi, yang dimaksud terlalu banyak informasi disini adalah banyaknya informasi yang berserakan belum terkumpul, belum terformat, dan masih tercampurnya antara informasi yang relevan dan yang tidak relevan dengan masalah yang harus diambil keputusannya, sehingga memerlukan waktu yang lebih lama untuk memilah dan memilih informasi yang relevan.

Informasi tidak dalam format yang bergun, adalah bahwa informasi sudah tersedia, hanya saja bentuk dan format nya tidak sesuai dengan yang dibutuhkan sehingga mempersulit pembaca informasi tersebut dan memerlukan waktu yang lebih lama untuk memahami dan memanfaatkan informasi tersebut.

3. *Economics*

Secara umum keuntungan-keuntungan yang didapat ketika menerapkan sistem informasi, selain yang tersebut dibawah ini masih ada lagi keuntungan-keuntungan yang lain yang secara lebih lengkap diidentifikasi. Sehingga pada bagian ini dideskripsikan manfaat yang akan didapatkan ketika menerapkan teknologi informasi atau sistem informasi dalam menjalankan proses bisnisnya.

4. *Control*

Pada bagian ini dideskripsikan situasi saat ini tentang kendali terhadap aliran data dan informasi ketika keamanan atau kendali terlihat lemah sehingga data dan informasi rentan terhadap pemanfaatan kepada pihak-pihak yang tidak berwenang. Juga ketika keamanan atau kendali terhadap aliran data dan informasi terlalu ketat sehingga sistem jadi terbebani oleh prosedur keamanan atau kendali tersebut dan juga mengganggu keamanan dan kenyamanan para pengguna dan pengambil manfaat data dan informasi yang dihasilkan oleh sistem tersebut.

5. *Efficiency*

Dimana data yang berlebihan diinputkan dan diproses juga informasi yang dihasilkan secara berlebihan akan membuat sistem tidak akan efisien dalam penggunaan sumber daya. Sumber daya dapat berupa sumber daya prosesor, memory, ruang penyimpanan, listrik, personil, dll.

6. *Service*

Pada bagian ini dideskripsikan situasi saat ini tentang layanan yang disediakan oleh sistem yang berjalan saat ini. Bagaimana pelayanan sistem yang diberikan saat ini.

2.4 Metode Pengolahan Data

Pengertian dari pengolahan data atau data processing merupakan manipulasi data ke bentuk yang lebih informative atau berupa informasi. Informasi merupakan hasil dari kegiatan pengolahan suatu data dalam bentuk tertentu yang lebih berarti dari suatu kegiatan atau suatu peristiwa.

Terdapat empat penggolongan alat pengolahan data yang bisa anda ketahui diantaranya peralatan manual atau alat sederhana untuk mengolah data dengan factor terpenting adalah dalam penggunaan alat dengan tenaga tangan manusia. Selanjutnya adalah peralatan mekanik yaitu peralatan yang sudah lebih modern dan dalam bentuk mekanik dan digerakkan oleh tangan manual.

Peralatan berikutnya adalah dengan peralatan mekanik elektronik yang digerakkan secara otomatis dengan motor elektronik. Dan alat yang terakhir adalah peralatan elektronik yang dikerjakan secara elektronik penuh tanpa bantuan tangan manusia.

2.4.1 Jenis data

Data menurut jenisnya ada dua yaitu data kualitatif dan data kuantitatif sebagai berikut:

1. Data kualitatif

Data yang berhubungan dengan kategorisasi, karakteristik berwujud pertanyaan atau berupa kata-kata. Contohnya wanita itu cantik, pria itu tampan, baik, buruk, rumah itu besar dan sebagainya. Data ini biasanya didapat dari wawancara yang bersifat subyektif sebab data tersebut ditafsirkan lain oleh orang yang berbeda. Data kualitatif dapat diangkakan dalam bentuk ordinal atau rangking.

2. Data kuantitatif

Data yang berwujud angka-angka. Contohnya ; yang diterima menjadi PNS 150 orang, penghasilan klinik bersalin 1 milyar bulan. Data ini diperoleh dari pengukuran langsung maupun dari angka-angka yang diperoleh dengan mengubah data kualitatif menjadi data kuantitatif. Data kuantitatif bersifat objektif dan bisa ditafsirkan oleh semua orang.

2.5 Pemodelan Berorientasi Objek

2.5.1 Pengertian Pemodelan Berorientasi Objek

Pendekatan berorientasi objek merupakan adalah suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi atau sistem lainnya). Sistem berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek – objek dunia nyata. Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahapan analisis, perancangan, pemrograman dan pengujian perangkat lunak. Ada berbagai teknik yang digunakan pada masing – masing tahapan tersebut, dengan aturan dan pemodelan tertentu. Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem dapat mewarisi sifat atribut dan sifat dari komponen lainnya dan dapat berinteraksi satu sama lainnya. Rosa A.S. dan M. Shalahuddin dalam bukunya *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (Rosa A.S. dan M. Shalahuddin, 2018).

2.5.2 Ciri – Ciri Pemodelan Berorientasi Objek

1. Kelas (*class*)

Kelas adalah kumpulan objek – objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi atau metode), hubungan (relationship) dan arti. Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode pemrogram yang merupakan metodologi berorientasi objek. Kelas secara fisik adalah berkas atau file yang berisi kode program, dimana kode program merupakan semua hal yang terkait dengan nama kelas. (Rosa A.S. dan M. Shalahuddin, 2018), OM Febriani, AS Putra, RP Prayogie - Prosiding Seminar Nasional Darmajaya, (2020).

2. Objek (*object*)

Objek adalah suatu yang mewakili dunia nyata seperti benda, manusia, tempat, struktur, status atau hal lainnya yang abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau berpengaruh pada suatu objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi maka akan dibuat sebuah objek. Objek dilihat dari segi teknis elemen pada saat runtime yang akan diciptakan, dimanipulasi dan dihancurkan saat dieksekusi maka sebuah objek hanya ada pada saat sebuah program dieksekusi. (Rosa A.S. dan M. Shalahuddin, 2018), AS Karim - JUPITER (Jurnal Penelitian Ilmu dan Teknik Komputer), (2017).

3. Metode (*method*)

Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode berasal dari event, aktivitas atau aksi keadaan, fungsi atau kelakuan dunia nyata. Contoh metode misalnya read, write, move, copy dan sebagainya. Kelas sebaiknya memiliki metode get dan set untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode get digunakan untuk memberi akses kelas lain dalam mengakses atribut, dan set adalah metode yang digunakan untuk mengisi atribut, agar kelas lain tidak mengakses atribut secara langsung. (Rosa A.S. dan M. Shalahuddin, 2018), D Handoko, SY Irianto, S Karnila - Journal of Physics: Conference Series, (2020).

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas

objek. Atribut dimiliki secara individu oleh sebuah objek. Atribut sebaiknya bersifat private untuk menjaga konsep enkapsulasi. (Rosa A.S. dan M. Shalahuddin, 2018)

5. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan pelayanan (perasi-operasi) yang dipunyai oleh objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya. (Rosa A.S. dan M. Shalahuddin, 2018).

6. Antarmuka (*interface*)

Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain. Metode pada antar muka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka. Antarmuka atau interface biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas tapi mengakses antarmukanya. (Rosa A.S. dan M. Shalahuddin, 2018)

7. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas – kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda. (Rosa A.S. dan M. Shalahuddin, 2018)




2.5.3 UML (*Unified Modeling Language*)

Menurut (Rosa & Shalahuddin, 2018) UML adalah: *Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

2.6.3.1 Use Case Diagram

Merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa A.S dan M. Shalahuddin, 2018).

Tabel 2.1 Simbol *Usecase Diagram* (Rosa A.S dan M. Shalahuddin, 2018)

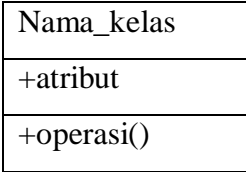



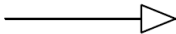
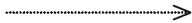

No	Simbol	Keterangan
1.	<i>UseCase</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	<i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<i>Association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada usecase atau use case memiliki interaksi dengan aktor.
4.	Ekstensi <i>Extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu.
5.	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	<i>Include</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

2.5.3.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. (Rosa A.S dan M. Shalahuddin, 2018).

Sebuah kelas diagram terdiri dari sejumlah kelas yang dihubungkan dengan garis yang menunjukkan hubungan antar kelas.

Tabel 2.2 Simbol *Class Diagram* (Rosa A.S dan M. Shalahuddin, 2018)


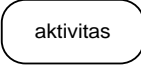



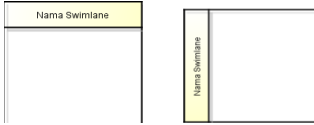
Simbol	Keterangan
KelasClass 	Kelas pada struktur system
Antar mukainterface 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
Asosiasiassociation 	Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarahdirected 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan atau <i>Dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasiaggregation 	Relasi antar kelas dengan makna semua-bagian(whole-part)

2.5.3.3 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan kator,jadi aktivitas yang dapat dilakukanoleh sistem. (Rosa A.S dan M. Shalahuddin, 2018)

Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Activity Diagram*(Rosa A.S dan M. Shalahuddin, 2018)



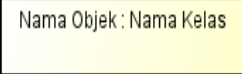

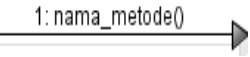
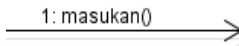
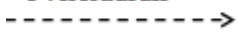
Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.5.3.4 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas. (Rosa A.S dan M. Shalahuddin, 2018)

Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.4 Simbol *Sequence Diagram* (Rosa A.S dan M. Shalahuddin, 2018)

Simbol	Keterangan
<p>Aktor</p>  <p>atau</p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatukan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe <i>create</i></p> <p><<create>></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirim data masukan informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.</p>
<p>Pesan Tipe <i>Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

2.6 Alat Implementasi Sistem

2.6.1 MySQL

(Loka Dwirtara, 2018) MySQL bersifat RDBMS (Relational Database Management Sistem) yang memungkinkan seorang admin dapat menyimpan banyak informasi ke tabel-tabel, dimana tabel-tabel tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah database ke dalam tabel-tabel yang berbeda, setiap tabel memiliki informasi yang berkaitan dengan tabel yang lainnya.

MySQL adalah *multi userdatabase* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan server daemon MySQL di sisi *server* dan sebagai macam program serta library yang berjalan di sisi client MySQL mampu menangani data yang cukup besar.

2.6.2. PHP

Menurut (Firmansyah, 2018) mendefinisikan bahwa PHP (*Hypertext Preprocessor*) adalah PHP mengizinkan pengembang untuk menempelkan kode di dalam HTML dengan menggunakan bahasa yang sama seperti perl dan UNIX shells. Sedangkan menurut (Isty & Afifah, 2018), PHP adalah pemrograman interpreter yaitu penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Menurut pendapat di atas dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan bersamaan dengan HTML dimana bahasa tersebut dijalankan di sisi *web browser*.

2.6.3. Xampp

Fungsi Xampp sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama Xampp sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya Xampp anda dapat mendownload langsung dari web resminya.

2.7 Metode Pengembangan Sistem

Metodologi pengembangan sistem pada penelitian ini menggunakan metode *Prototype*. Menurut (Adi Nugroho, 2018) metode *prototype* dibuat saat pengguna tidak tahu pasti apa yang mereka inginkan baik rincian masukannya, rincian proses dan rincian keluaran yang diinginkan untuk itu dibuatlah *Prototype* kepada pengguna. Kemudian pengguna menyarankan perbaikan-

perbaikan jika terdapat kekurangan sistem yang perlu diperbaiki. Adapun tahapan-tahapannya adalah sebagai berikut :

1. Interaksi dengan pengguna

Pada tahapan ini penyusun menganalisis apa yang ingin pengguna dapatkan dari sistem/perangkat lunak itu. Sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan sistem.

2. Membuat *Prototype*

Pada tahapan ini akan dibuat sebuah *Prototype* aplikasi berbasis web berdasarkan atas kebutuhan pengguna dan sistem pada tahap interaksi dengan pengguna.

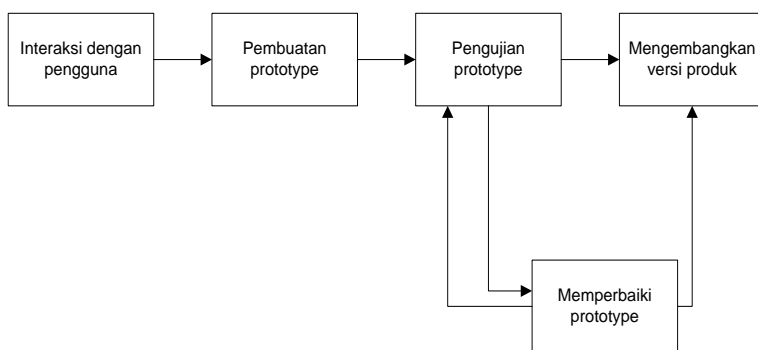
3. Menguji *Prototype*

Tahapan ini adalah proses penilaian terhadap *Prototype* yang telah dibuat apakah sesuai dengan kebutuhan atau tidak jika tidak maka *Prototype* akan diperbaiki.

4. Memperbaiki *Prototype*

Setelah ditemukan letak kesalahan dari *Prototype* yang dirancang pada tahapan ini penyusun akan membuat atau memperbaiki *Prototype* yang ada setelah itu akan di uji lagi sehingga *Prototype* sesuai dengan keinginan pengguna.

5. Mengembangkan Versi Produk setelah aplikasi dapat berjalan dan memenuhi kebutuhan sistem maka aplikasi ini siap dipakai Metode *Prototype* ini dapat digambarkan sebagai berikut :



Gambar 2.1 Metode *Prototype*

Prototype memiliki tahapan tahapan sebagai berikut :

a. Identifikasi kebutuhan *end* Penduduk

Pada tahap ini para pemakai akhir mengidentifikasi kebutuhan bisnis mereka dan menilai kelayakan beberapa alternatif solusi sistem informasi. Pengguna sistem informasi dan

vendor mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat beserta alternatif solusi sistem.

b. Membangun *prototype business system*

Pada tahap ini para pemakai akhir atau pakar sistem informasi menggunakan alat pengembangan aplikasi untuk secara interaktif mendesain dan menguji *prototype* berbagai komponen sistem informasi yang memenuhi kebutuhan para pemakai akhir. Membangun *prototyping* aplikasi pengembangan dengan membuat model sebagai uji coba yang mewakili kebutuhan pengguna secara garis besar.

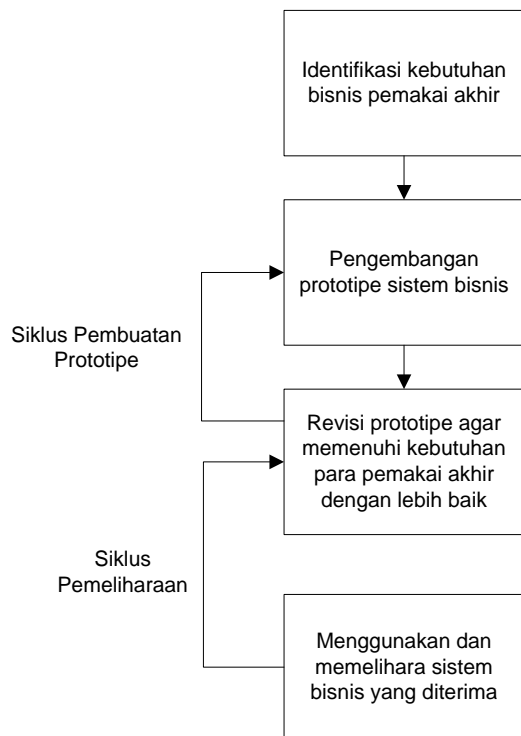
c. Revisi *prototype* kedalam bentuk yang mendekati kebutuhan *end* Penduduk

Model sistem bisnis diuji coba, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima oleh pengguna dan dirasakan oleh pengguna telah sesuai dengan kebutuhan.

d. Menggunakan dan memelihara *business system* yang telah diterima

Dalam tahap ini sistem bisnis yang telah disepakati antara pengguna dan *vendor* dapat dimodifikasi dengan mudah karena sebagian besar dokumentasi dari sistem telah tersimpan.

Model *prototype* dimulai dengan pengumpulan kebutuhan. Pengembang dan Penduduk bertemu dan mendefinisikan obyektif keseluruhan dari *software*, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perencanaan secara cepat. Sehingga *prototyping* dikenal dengan *Rapid Application Development* (RAD). Skema tahapan pengembangan sistem informasi dengan pendekatan *prototyping* menurut O'Brien :



Gambar 2.2 Skema tahapan pengembangan sistem informasi dengan pendekatan *prototyping*

2.8 Penelitian Sebelumnya

Tabel 2.4 Tabel Referensi Penelitian Terkait

No	Nama Peneliti	Judul Penelitian	Tahun	Metode	Hasil
1	Destina Sari, Endang Masitoh W	Evaluasi Sistem Informasi Akuntansi Atas Prosedur Penerimaan & Pengeluaran Dana Bantuan Operasional Sekolah	2018	<i>UML</i>	Penelitian ini bertujuan mengevaluasi Sistem Informasi Akuntansi atas prosedur penerimaan dan pengeluaran dana BOS pada SMP Negeri 2 Andong. Populasi dalam penelitian ini adalah Prosedur

		(Bos) Pada Sekolah Menengah Pertama Negeri 2 Andong Boyolali			Penerimaan dan Pengeluaran dana Bantuan Operasional Sekolah
2	Wiwit, Tri Sulistyaningsih, dan Muhammad Kamil	Monitoring dan Evaluasi Kebijakan Program Bantuan Pangan Non Tunai (BPNT) Dalam Penanggulangan Kemiskinan di Kota Batu	2020	UML	bertujuan untuk menjelaskan Permasalahan pada program pemerintah dalam pemberian bantuan sosial pada masyarakat miskin yang belum tepat. Berdasarkan arahan presiden Joko Widodo melalui Ratas pada 16 Maret 2016 menyatakan agar bantuan sosial diberikan secara non tunai, salah satunya melalui program BPNT berdasarkan Permensos No.11 Tahun 2018 dimana merupakan transformasi dari program Rastra

					yang memiliki masalah dalam penyalurannya
3	Teguh Hamdala, Shinta Esabella	Rancang Bangun Aplikasi Monitoring Dan Evaluasi Penyaluran Dana Sosial Di Dinas Sosial Kabupaten Sumbawa Berbasis Web	2020	UML	Penelitian Ini Bertujuan Untuk Merancang Dan Membangun Aplikasi Monitoring Dan Evaluasi (Monev) Penyaluran Dana Sosial Di Dinas Sosial Kabupaten Sumbawa Berbasis Web Agar Dapat Membantu Dinas Sosial Di Bidang Fasilitasi Perlindungan Dan Jaminan Sosial Dalam Melaporkan Atau Evaluasi Terhadap Kondisi Dana Sosial Sekaligus Juga Melakukan Proses Monitoring Perubahan Penerima Dana Sosial Di Kabupaten Sumbawa

