

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem didefinisikan sebagai seperangkat komponen yang saling terhubung, dengan sebuah batasan yang jelas, bekerjasama untuk mencapai sebuah tujuan yang sama dengan menerima masukan dan menghasilkan keluaran dalam sebuah proses transformasi yang terorganisasi [1].

Sistem adalah sekelompok komponen-komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur.

Sistem semacam ini (kadang disebut sebagai sistem dinamis) memiliki tiga komponen atau fungsi dasar yang berinteraksi:

- a. *Input*, melibatkan penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (permintaan jasa dari pelanggan).
- b. Pemrosesan, melibatkan proses transformasi yang mengubah *input* menjadi *output*. Keluaran yang dihasilkan dapat berupa informasi maupun produk. Pada sistem informasi, proses dapat berupa meringkas data, melakukan perhitungan dan mengurutkan data.
- c. *Output*, melibatkan perpindahan elemen yang telah diproduksi oleh proses transformasi ke tujuan akhirnya. Pada sistem informasi, keluaran dapat berupa informasi, saran, cetakan, laporan dan sebagainya [2]

2.1.2 Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata [3].

Informasi dapat berupa dokumen operasional seperti pesanan penjualan, laporan yang terstruktur, atau sebuah pesan dalam layar komputer. Adapun bentuk fisiknya, informasi yang berguna memiliki berbagai karakteristik berikut ini:

- a. Relevan, isi dari suatu laporan atau dokumen harus berkerja untuk suatu tujuan. Ini dapat berupa dukungan bagi keputusan manajer atau untuk pekerja staf administrasi. Kita telah menetapkan bahwa data yang relevan dengan tindakan penggunaannya yang memiliki nilai informasi.
- b. Tepat waktu, umur informasi adalah faktor yang sangat penting dalam menentukan kegunaannya. Informasi harus tidak melebihi periode waktu dari tindakan yang didukungnya.
- c. Akurasi, informasi harus bebas dalam kesalahan yang signifikan. Akan tetapi, signifikansi adalah konsep yang sulit untuk diukur. Konsep ini tidak memiliki nilai absolut.
- d. Kelengkapan, semua informasi yang penting bagi keputusan atau pekerjaan harus ada. Contohnya, sebuah laporan harus menyediakan semua perhitungan yang dibutuhkan dan menyajikan pesanan secara jelas serta tidak ambigu [4]

2.1.3 Sistem Informasi

Sistem informasi merupakan suatu kumpulan elemen yang saling bekerjasama untuk mempermudah suatu proses dan menghasilkan suatu informasi yang bernilai. Teknologi informasi saat ini tidak luput dari penggunaan internet. Internet telah menjadi sistem komunikasi dunia yang paling luas dan umum yang sekarang menyaingi sistem telepon global dalam jangkauan dan jarak [5].

Sistem informasi dapat merupakan kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam suatu organisasi. Orang bergantung pada sistem informasi untuk berkomunikasi antara satu sama lain dengan menggunakan berbagai jenis alat fisik (*hardware*), perintah dan prosedur pemrosesan informasi (*software*), saluran komunikasi (jaringan) dan data yang disimpan (sumber daya data) sejak mula peradaban [2]

2.2 Website

Website adalah kumpulan dari halaman-halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada didalam *World Wide Wide* (WWW) di dalam internet. Sebuah halaman *web* biasanya berupa dokumen yang ditulis dalam format HTML (*Hyperteks Markup Language*), yang selalu bisa diakses melalui HTTP (*HyperText Trasfer Protocol*), yaitu sebuah protokol yang menyampaikan informasi dari server *website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar. Halaman-halaman dari *website* dapat diakses melalui sebuah URL (*Uniform Resource Locator*) yang biasanya disebut *homepage*. URL ini mengatur halaman situs untuk menjadi sebuah piranti, meskipun *hyperlink* yang ada dihalaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa *website* membutuhkan subskripsi (data masukan) agar para *user* bisa mengakses sebagian atau keseluruhan isi *website* tersebut. Contohnya ada beberapa situs-situs bisnis, situs-situs *email gratis*, yang membutuhkan subskripsi agar kita dapat mengakses situs tersebut [6]

Terdapat dua jenis *website*, yaitu *website* statis dan *website* dinamis. *Website* statis adalah salah satu bentuk *website* yang isi didalam *website* tersebut tidak dimaksudkan untuk di *update* secara berkala, dan biasanya di *maintenance* secara manual oleh beberapa orang yang menggunakan *software editor*. Sedangkan *website* dinamis adalah *website* yang secara berkala, informasi didalamnya berubah, atau *website* ini bisa berhubungan dengan *user* dengan berbagai macam cara atau metode (HTTP *Cookies* atau Variabel *Database*, sejarah kunjungan, Variabel Sesi dan lain-lain) bisa juga dengan cara interaksi langsung menggunakan *form* dan pergerakan *mouse* [7].

Sekarang ini ada banyak sekali jenis *website* yang bisa kita temukan di internet. Namun, semua *website* tersebut dapat kita klasifikasikan ke dalam tiga jenis, yaitu:

1. *Website* Statis

Pengertian *website* statis adalah suatu halaman *website* yang tampilannya tidak berubah-ubah (statis). Jika si pemilik *website* ingin mengubah tampilan maka harus dilakukan secara manual, yaitu dengan mengedit kode-kode struktur

websitenya. Jenis *website* statis umumnya memiliki setidaknya 5 halaman utama untuk menjelaskan informasi mengenai *website* tersebut. Selain itu, *website* statis umumnya tidak memungkinkan terjadinya interaksi langsung antara pemilik atau pengelola *website* dan pengunjung di *website* tersebut.

Contoh *website* statis:

- *Website* perusahaan (*company profile*)
- *Search Engine* (*Google, Bing*)

2. *Website* Dinamis

Website dinamis adalah jenis *website* yang dirancang khusus untuk dapat menampilkan *update* konten sesering mungkin. *Website* dinamis dapat disesuaikan dengan kebutuhan, baik dari sisi tampilannya maupun dari sisi fiturnya. *Website* dinamis umumnya dirancang dengan konsep visual dan kemampuan interaksi tinggi dengan penggunaanya. Beberapa fitur yang bisa ditambahkan pada *website* dinamis yaitu, kolom komentar, *fitur live chatting*, formulir pendaftaran, dan lain-lain.

Beberapa contoh *website* dinamis:

- *Blog* atau *website* pribadi
- Situs *E-commerce*
- *Website* Portal
- Situs Berita
- Katalog *online*

3. *Website* Interaktif

Website interaktif adalah jenis *website* yang digunakan untuk tujuan berinteraksi dengan orang lain secara *online*. Umumnya pengguna *website* interaktif adalah komunitas atau pengguna internet aktif.

Beberapa contoh *website* interaktif:

- Situs Media sosial
- Situs forum *online*
- *Blog*

Manfaat *website* yang paling utama adalah penyebaran informasi yang lebih cepat kepada masyarakat luas. Internet dan *website* berperang penting dalam proses pertukaran informasi, baik secara lokal maupun internasional.

Berikut ini adalah beberapa manfaat *website* secara umum:

- Sebagai sarana informasi yang lebih mudah dan cepat untuk didapatkan
- Memberikan kemudahan dalam kegiatan pemasaran dan promosi bisnis karena dapat menjangkau banyak orang dalam waktu yang bersamaan
- *Website* menjadi sarana berkomunikasi bagi manusia di berbagai belahan dunia
- Sebagai sarana edukasi bagi masyarakat Sebagai sarana hiburan yang murah

2.3 Use Case

2.3.1 Use Case Diagram

Diagram *use-case* (*use-case diagram*) merupakan suatu bentuk *diagram* yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem. Mempresentasikan interaksi yang terjadi antara aktor dengan proses atau sistem yang dibuat. Interaksi antara *use case* dengan asosiasi berupa garis sederhana. Sebuah *use case* dapat berinteraksi dengan lebih dari satu *actor* dan dapat berinteraksi lebih dari satu *use case* [8].

Tujuan penggunaan *use case diagram*:

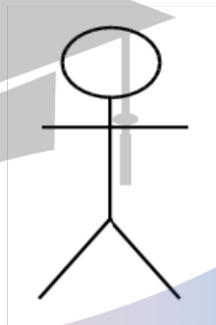
1. Mendapatkan pemahaman tentang sistem/perangkat lunak yang akan dikembangkan,
2. Memperlihatkan hubungan-hubungan yang terjadi antara *actor* (seseorang/sesuatu yang berinteraksi dengan sistem) dengan *use case* (proses yang terjadi dalam sistem).
3. Membantu dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem,
4. Dengan melihat aktor-aktor, pengguna akan mengetahui siapa atau apa saja yang akan berinteraksi dengan sistem.

5. Dengan melihat kombinasi sejumlah *actor* dan *use case*, pengguna akan mengetahui secara jelas ruang lingkup dari sistem/perangkat lunak yang akan dikembangkan [8] .

Komponen pada *use case* terdiri dari:

1. *Actor*

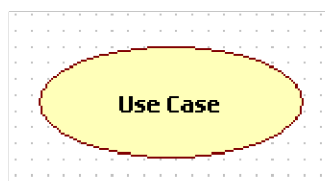
Aktor (*actor*) adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor mempresentasikan peran bukan pemakai individu dari sistem. Aktor mempunyai nama dan nama yang dipilih seharusnya menyatakan peran aktor.



Gambar 2.1 *Actor* [8]

2. *Use Case*

Use Case adalah cara spesifik pengguna sistem oleh *actor*. *Use case* menspesifikasikan perilaku sistem atau bagian sistem dan merupakan deskripsi sekumpulan sekuen aksi termasuk varian-varian yang dilakukan sistem untuk memproduksi hasil atau nilai ke aktor.

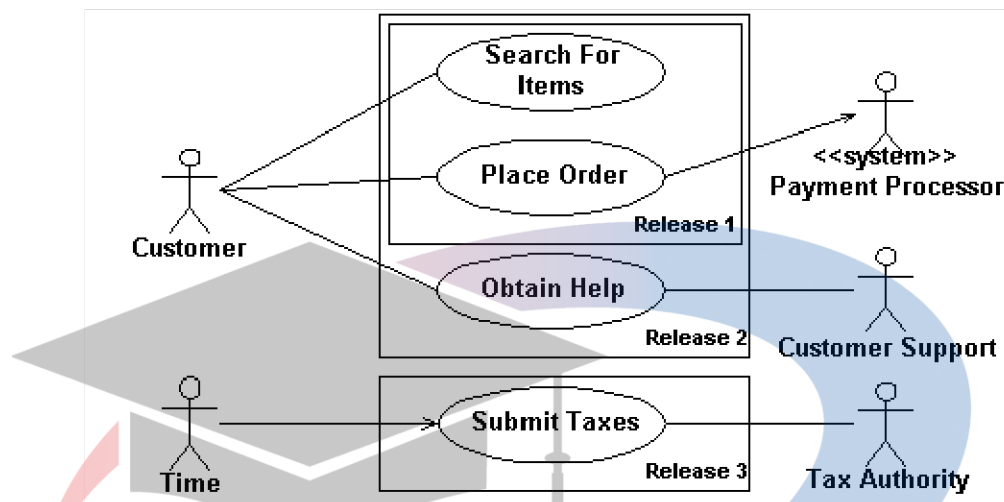


Gambar 2.1 *Use Case* [8]

3. *System Boundary*

System boundary merupakan ruang lingkup sistem yang berisi *use case*, *use case* yang merupakan fungsional dari sistem tersebut dan antara sistem yang satu

dengan sistem yang lain dimungkinkan saling berinteraksi. *System Boundary* dilambangkan dengan sebuah persegi panjang.



Gambar 2.2 Sistem Boundary [8]

4. Association

Association merupakan arah panah yang menunjukkan komunikasi antara *actor* dengan *use case* dan siapa yang mengawali komunikasi. *Association* digambarkan dengan sebuah garis sebagai relasi asosiasi.

- Sebaiknya gunakan garis tanpa panah untuk *association* antara *actor* dan *use case*.



Gambar 2.3 Association Tanpa Panah [8]

- Association* antara *actor* dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila *actor* berinteraksi secara pasif dengan sistem.



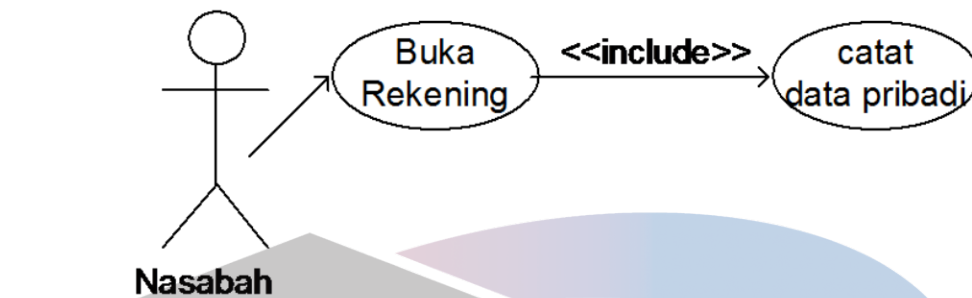
Gambar 2.4 Association Dengan Panah [8]

Pada *association* terdapat 4 jenis relasi yang bisa timbul pada *use case diagram*, yaitu:

- Association* antara *actor* dan *use case*

Include termasuk didalam *use case* lain (*required*) / (diharuskan).

Pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program. Tanda panah terbuka harus terarah ke *sub use case*.

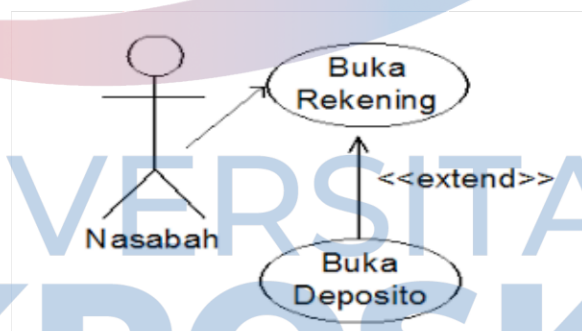


Gambar 2.5 Association Antara Actor dan Use Case [8]

2. Association antara use case

Extend perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

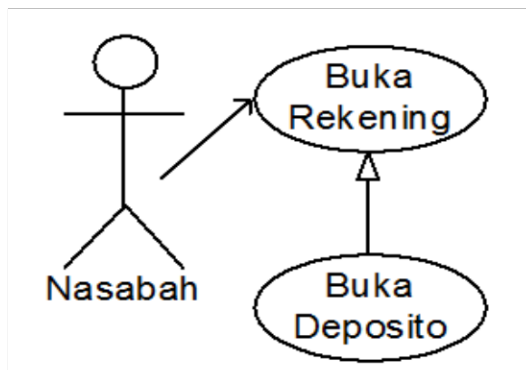
Pemakaian *association* ini membuat diagram sulit dipahami dan tanda panah terbuka harus terarah ke *parent/base use case*.



Gambar 2.6 Association Antara Use Case [8]

3. Generalization/ Inheritance antara use case

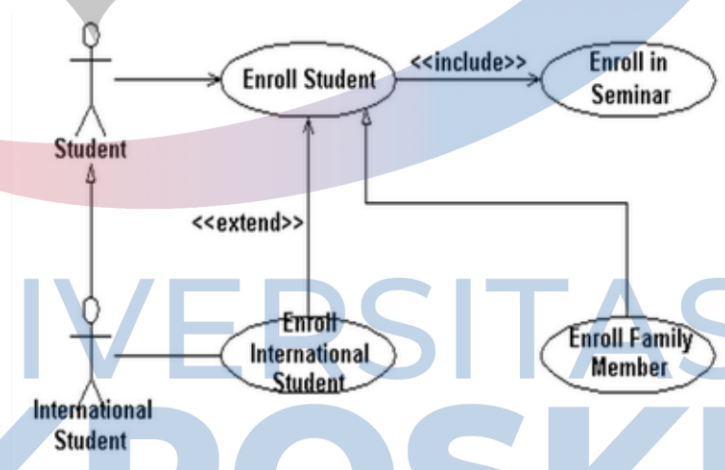
Generalization/ Inheritance digambarkan dengan sebuah garis panah tertutup pada salah satu ujungnya yang menunjukkan lebih umum. *Generalization/ Inheritance* dipakai ketika ada sebuah keadaan yang lain sendiri/perlakuan khusus (*single condition*).



Gambar 2.7 Generalization/ Inheritance Antara Use Case [8]

4. Generalization/ Inheritance antara actor

Generalization/ Inheritance antara actor secara vertical dengan actor dibawah base/ parent use case.



Gambar 2.8 Generalization/ Inheritance antara actor. [8]

2.3.2 Use Case Description

Use Case Description adalah salah satu dari *diagram* UML yang bertujuan untuk memberikan gambaran umum tentang fungsionalitas suatu proses bisnis yang didalamnya melibatkan sebuah *system* [9].

Di dalam *Use Case Description* sendiri terdapat beberapa elemen-elemen antara lain:

1. Overview Information

Elemen ini menyediakan *basic information* dari sebuah *use case*. Dimana informasi yang di berikan mencakup:

- *Use case name*: Berisikan informasi tentang nama *use case* yang akan didetailkan.
- *Use case ID*: Berisikan kode unik dari tiap *use case* yang ada, biasanya merupakan urutan dari *use case*.
- *Importance level*: Berisikan informasi tentang seberapa penting atau *Critical*, *use case* yang dibuat.
- *Primary actor*: Berisikan informasi tentang pengguna atau *actor* yang menjalankan *use case* tersebut.
- *Use case type*: Berisikan informasi tentang tipe dari tiap *use case*.
- *Stakeholder*: Berisikan informasi tentang seseorang yang nantinya Berkepentingan terhadap *use case* yang dibuat.
- *Brief description*: Berisikan informasi tentang *use case* secara umum.
- *Trigger*: Berisikan informasi tentang sebuah *event* yang memicu dibuatnya *use case*.
- *Trigger type*: Berisikan informasi tentang tipe dari pemicu dari *use case* itu terjadi, dimana terdapat dua macam tipe pemicu yaitu *External trigger* dan *temporal trigger*. Pemicu *External* adalah kejadian *external* yang memicu sebuah *use case* dibuat sedangkan pemicu *Temporal* adalah suatu jangka waktu yang nantinya memicu sebuah *use case* dibuat.

2. Relationship

Elemen ini menjelaskan bagaimana hubungan sebuah *use case* dengan *use case* yang lain maupun hubungan antara *use case* dengan para *actornya*.

Ada 4 *basic type* dari *relationship* di dalam sebuah *use case* antara lain:

- *Association relationship*: Menjelaskan hubungan antara *use case* dan *actor* yang terlibat di dalam *use case* tersebut.
- *Include relationship*: Menjelaskan *inclusion* dari tiap *use case*, yang dimana *Include* ini bertujuan untuk memecah sebuah *use case* yang kompleks menjadi beberapa bagian terpisah.

- *Extend relationship*: Menjelaskan tentang ekstensi dari tiap *use case* yang ada, biasanya ekstensi dari *use case* ini tidak bisa dijalankan jika *use case* utamanya tidak dijalankan terlebih dahulu.
- *Generalization relationship*: Menjelaskan tentang *Generalization* dari tiap *use case*, Seperti contohnya membuat *order* di kelompokkan atau dibagi menjadi membuat *order member* lama dan membuat *order member* baru.

3. *Flow of event*

Elemen ini adalah elemen terakhir dalam penyusunan *Use case description* yang dimana elemen ini menjelaskan tentang proses bisnis dari tiap *use case* yang ada. *Flow of event* sendiri dibagi menjadi tiga *steps* antara lain:

- *Normal flow*: Berisikan tentang *steps* atau langkah langkah umum dari proses bisnis yang ada dari tiap *use case*.
- *Sub-Flow*: Berisikan tentang proses bisnis yang dipecah dari *normal flow* yang terlalu kompleks sehingga menjadi sebuah proses lain yang masih berhubungan dengan *normal flow*nya.
- *Alternative/ Exceptional flow*: Berisikan informasi tentang kendala-kendala yang menyebabkan proses bisnis tidak dapat berjalan secara Baik.

Dari tiap tiap *use case* yang ada kita deskripsikan secara detail dengan *Use Case Description*.

- Tahap awal untuk membuat *Use Case Description* adalah dengan membuat tabel *Use Case Description*.
- Tahap Kedua adalah *men-define*:
 1. *Overview Information*
 2. *Relationship*
 3. *Flow of Event*

2.4 Prototyping

Prototype adalah satu versi dari sebuah sistem potensial yang memberikan ide dari para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. [10]

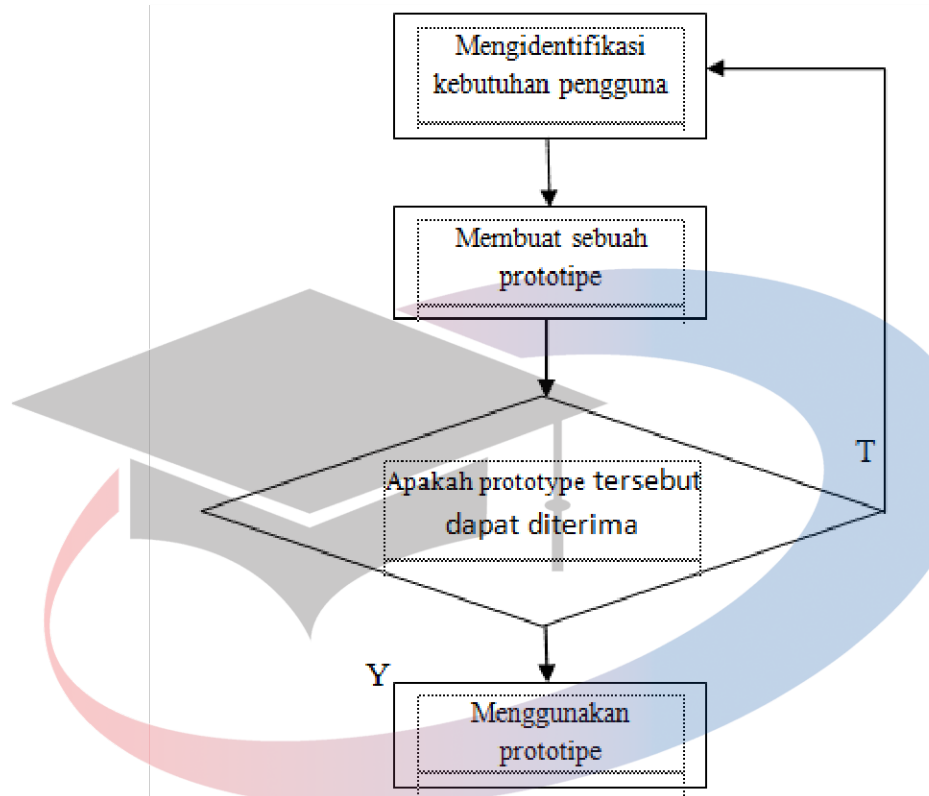
Proses pembuatan *prototype* ini disebut *prototyping*. Dasar pemikirannya adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam, lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat. Terdapat dua jenis *prototype* evolusioner dan persyaratan yaitu: [10]

1. *Prototype* evolusioner (*evolutionary prototype*) terus-menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. *Prototype* ini kemudian dilanjutkan produksi. Jadi satu *prototype* evolusioner akan menjadi sistem aktual.
2. *Prototype* persyaratan (*requirement prototype*) dikembangkan sebagai satu cara untuk mendefinisikan persyaratan-persyaratan fungsional dari sistem baru ketika pengguna tidak mampu mengungkapkan apa yang diinginkan. Dengan meninjau *prototype* persyaratan seiring dengan ditambahkannya fitur-fitur, pengguna akan mampu mendefinisikan pemrosesan yang dibutuhkan dari sistem yang baru. Ketika persyaratan ditentukan, *prototype* persyaratan telah mencapai tujuan dan proyek lain akan dimulai untuk pengembangan sistem baru.

Empat langkah dalam pembuatan suatu *prototype* evolusioner, yaitu: [10]

- a. Mengidentifikasi kebutuhan pengguna. Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
- b. Membuat satu *prototype*. Pengembang mempergunakan satu alat *prototyping* atau lebih untuk membuat *prototype*.
- c. Menentukan apakah *prototype* dapat diterima, pengembang mendemonstrasikan *prototype* kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan, jika sudah, langkah keempat akan diambil; jika tidak, *prototype* direvisi dengan mengulang langkah satu, dua dan tiga dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.

d. Menggunakan *prototype*, *prototype* menjadi sistem produksi.



Gambar 2.10 Alur Pembuatan *Prototype* [10]

UNIVERSITAS
MIKROSKIL