

BY : TSUKI~AMAI

MATERI UNTUK UJIAN TAP (Tugas Akhir Program)

1. DFD (Diagram Konteks/DFD Level 0, Level 1, Level 2)
2. ERD
3. SDLC (Metode)
4. Latar Belakang masalah, rumusan masalah, tujuan penelitian.




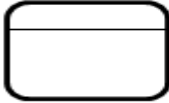




PEMBAHASAN 1

DFD digunakan untuk menggambarkan secara fisik dan tanpa mempertimbangkan fisik dimana data tersebut mengalir.

Komponen DFD :

- a. Proses merupakan suatu kegiatan / fungsi yang dilakukan dalam proses bisnis tertentu.
- b. Aliran data merupakan bagan data kadang disebut elemen data/ kumpulan beberapa informasi. Aliran data juga disebut penghubung yang menyatukan proses dan menunjukkan input apa yang akan masuk ke setiap proses dan output apa yang dihasilkan proses.
- c. Penyimpanan data merupakan kumpulan data yang disimpan dalam beberapa cara(ditentukan saat membuat model fisik). Penyimpanan data harus memiliki setidaknya satu aliran data input dan satu aliran data keluar.
- d. Entitas luar merupakan orang/sekelompok, organisasi/unit di luar organisasi atau sistem yang berada di luar sistem. Entitas luar menyediakan data ke sistem/menerima data dari sistem dan berfungsi menetapkan batas-batas sistem.

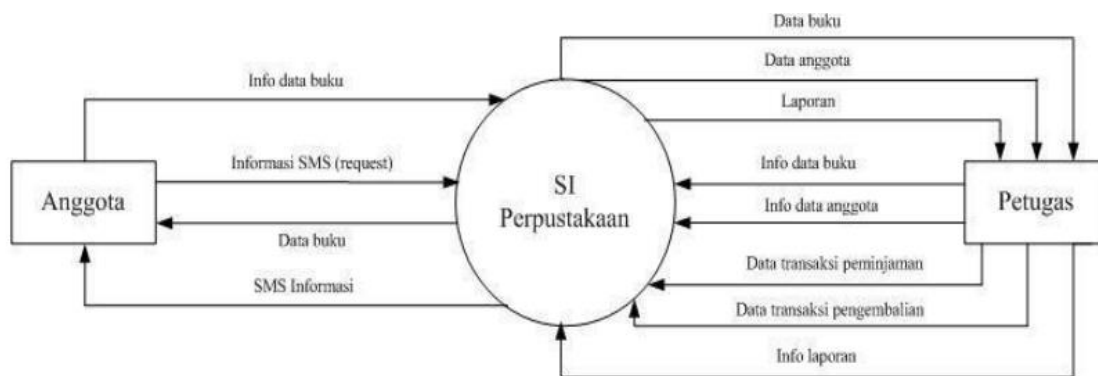
Simbol dari komponen DFD :

Keterangan	DeMarco and Yourdan Simbol	Gane and Sarson Simbol
Entitas Luar		
Proses		
Aliran data (data flow)		
Simpan data		

Jenis-jenis DFD

- Diagram Konteks menunjukkan keseluruhan proses bisnis hanya sebagai satu sistem (yaitu sistem itu sendiri dan menunjukkan aliran data ke dan dari entitas luar. Penyimpanan data biasanya tidak termasuk dalam diagram konteks, kecuali jika penyimpanan data itu dimiliki sistem lain.
- DFD level 1 merupakan lanjutan dari diagram konteks karena setiap proses yang berjalan akan diperinci pada tingkatan ini sehingga proses utama akan dipecah menjadi sub-sub proses yang lebih kecil lagi.
- DFD level 2 merupakan tingkat lanjutan dari level yang sebelumnya, dimana pada fase ini akan dijelaskan lebih detail terkait tiap prosesnya. Namun, untuk tingkatan ini jarang sekali dikerjakan dan lebih banyak hanya menerapkan dua level di bawahnya saja.

Gambar Diagram Konteks



Note : Gambar Contoh DFD Paling Bawah

PEMBAHASAN 2

ERD (Entity Relationship Diagram) merupakan model data konseptual yang mempresentasikan data dalam suatu organisasi dalam bentuk objek-objek dan hubungannya. ERD (Entity Relationship Diagram) atau diagram hubungan entitas adalah sebuah diagram yang digunakan untuk perancangan suatu database dan menunjukkan relasi atau hubungan antar objek atau entitas beserta atribut-atributnya secara detail.

Komponen-komponen ERD :

- Entitas** merupakan sekumpulan objek yang dapat diidentifikasi secara unik dan berbeda satu dengan yang lainnya. Entitas ini biasanya digambarkan dengan lambang persegi panjang.
- Atribut** yaitu Setiap entitas pasti memiliki atribut yang berfungsi untuk menjelaskan atau mendeskripsikan karakteristik dari entitas tersebut. Ada beberapa jenis atribut yang biasa digunakan dalam ERD. Berikut adalah jenis-jenisnya.

- **Atribut kunci**

Atribut kunci atau Key Attributes adalah atribut yang berfungsi untuk menentukan data yang bersifat penting. Biasanya atribut kunci ini berbentuk angka atau numerik. Contoh dari atribut ini adalah No. KTP, NIM (Nomor Induk Mahasiswa), dan lain-lain. Atribut kunci ini dilambangkan dengan lingkaran lonjong dengan keterangan di dalamnya yang diberi garis bawah.

- **Atribut simpel**

Berikutnya adalah atribut simpel. Atribut simpel adalah atribut yang tidak dapat dipecah lagi dan bernilai tunggal. Contoh dari atribut ini adalah alamat kantor, nama penerbit, dan lain-lain.

- **Atribut multinilai**

Atribut multinilai atau Multivalue Attributes adalah atribut yang memiliki atribut lebih dari satu nilai. Contoh dari atribut ini adalah sebuah website artikel yang memiliki beberapa penulis.

- **Atribut gabungan**

Selanjutnya adalah atribut gabungan atau Composite Attributes. Atribut gabungan adalah atribut yang terdiri dari beberapa atribut yang berukuran lebih kecil dan memiliki arti tertentu. Contoh dari atribut ini adalah sebuah nama yang terdiri atas nama depan, nama tengah, dan nama belakang.

- **Atribut derivatif**

Yang terakhir adalah atribut derivatif. Atribut derivatif adalah atribut yang dihasilkan dari atribut lain dan atributnya tidak wajib untuk ditulis dalam Entity Relationship Diagram. Contoh dari atribut ini adalah selisih harga, usia, dan kelas.

c. **Relasi**

Komponen ketiga adalah relasi atau relation. Relasi dalam ERD adalah hubungan yang terjadi antara satu atau lebih entitas. Relasi sendiri sering disebut dengan proses. Komponen ini digambarkan dengan lambang belah ketupat. Terdapat tiga jenis relasi yang digunakan dalam ERD dan perlu kamu ketahui, berikut adalah jenisnya.

- **One to one**

One to one berarti setiap entitas hanya dapat memiliki relasi dengan satu entitas lain. Contohnya seperti data mahasiswa dengan NIM (Nomor Induk Siswa).

- **One to many**

One to many memiliki arti satu entitas dapat memiliki relasi dengan beberapa entitas, begitu pula sebaliknya. Contoh dari implementasi one to many ini adalah jurusan dengan mahasiswanya.

- **Many to many**

Many to many memiliki arti setiap entitas yang ada dapat memiliki relasi dengan entitas lain, begitu pula sebaliknya. Contoh dari relasi ini adalah mahasiswa dengan data terkait UKM (Unit Kegiatan Mahasiswa).

- **Many to one (Banyak ke satu)** contohnya : mahasiswa diajar oleh 1 dosen.

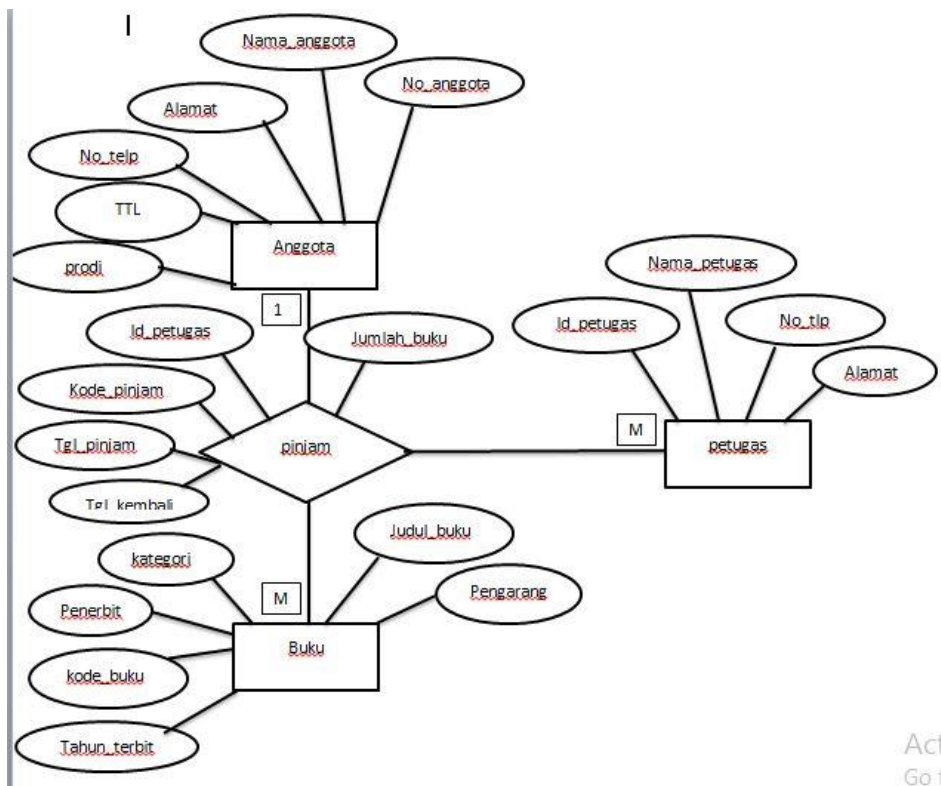
d. **Garis**

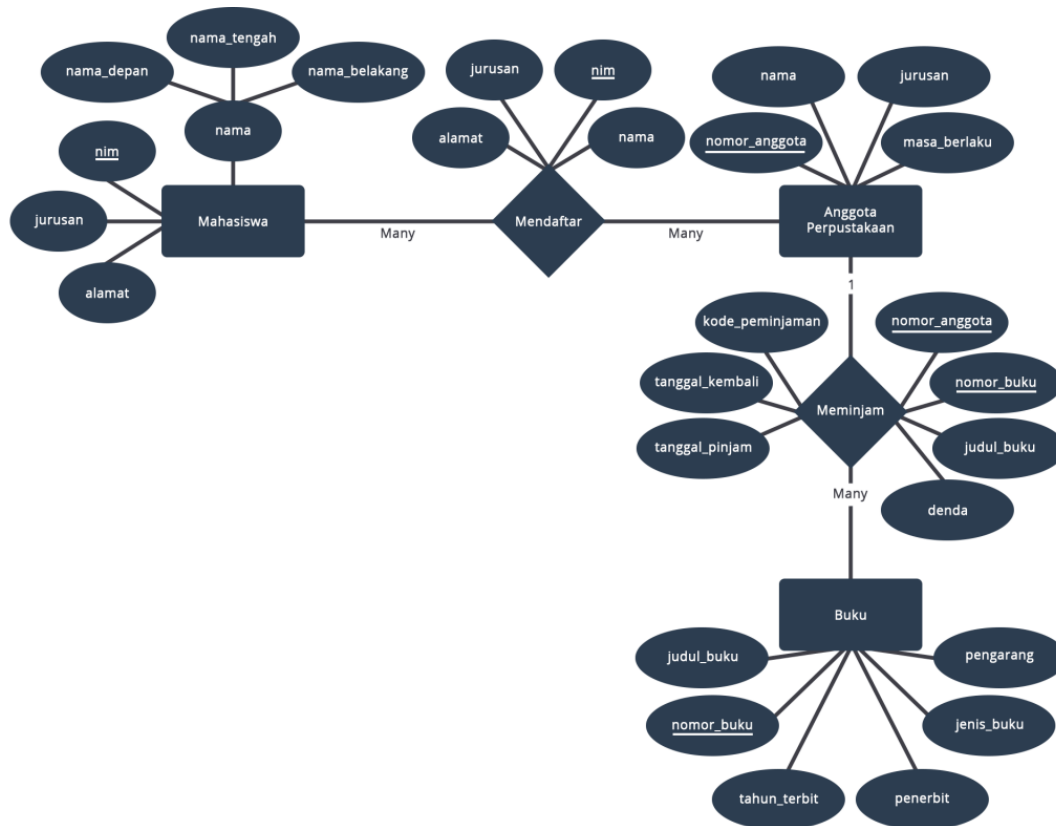
Komponen terakhir adalah garis. Dalam ERD sendiri garis digunakan untuk menunjukkan hubungan entitas dalam ERD. Selain menjadi penghubung, garis juga dapat menunjukkan alur atau flow dari suatu ERD.

Simbol-simbol dari ERD :

Notasi	Keterangan
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto;"></div> <p>Entitas</p>	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
<div style="border: 1px solid black; width: 50px; height: 30px; margin: 0 auto; transform: rotate(45deg);"></div> <p>Relasi</p>	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto; border-radius: 10px;"></div> <p>Atribut</p>	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
<div style="border: 1px solid black; width: 50px; height: 10px; margin: 0 auto;"></div> <p>Garis</p>	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.

Contoh ERD :





PEMBAHASAN 3

SLDC digunakan untuk membangun suatu sistem informasi agar dapat berjalan sesuai dengan apa yang diharapkan. Metode SLDC mampu menampilkan informasi dengan tepat, akurat dan berguna dalam membantu pengambilan keputusan dalam suatu organisasi.

Tujuan SLDC adalah meminimalkan risiko proyek melalui perencanaan ke depan agar perangkat lunak memenuhi harapan pelanggan selama produksi dan seterusnya.

Metode-metode SLDC :

1. **Waterfall/Air terjun** adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Disebut waterfall/ air terjun karena proses mengalir satu arah “ke bawah” seperti air terjun. Kelebihan menggunakan metode waterfall adalah memungkinkan untuk departemenisasi dan kontrol, proses pengembangan model fase one by one, sehingga meminimalis kesalahan yang mungkin terjadi.

Metode waterfall sangat cocok digunakan untuk kebutuhan pelanggan yang sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil.

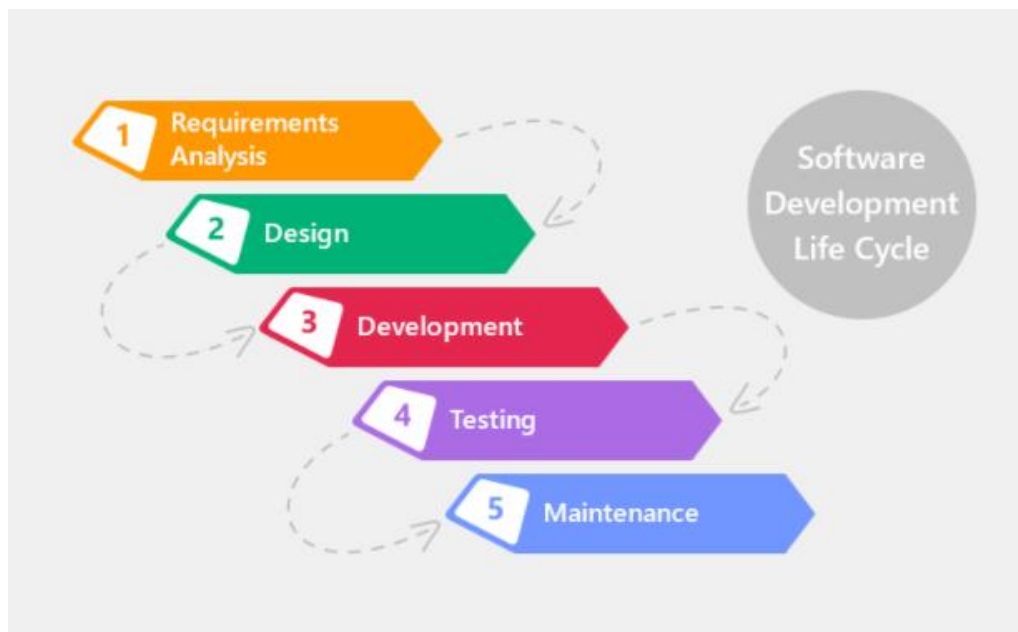
Tahapan metode waterfall yaitu :

- **Analisis kebutuhan perangkat lunak.** Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei,

wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

- **Desain (System and Software Design).** Perancangan desain dilakukan dengan tujuan membantu memberikan gambaran lengkap mengenai apa yang harus dikerjakan. Tahap ini juga akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.
- **Pembuatan Kode Program/implementasi.** Tahap implementation and unit testing merupakan tahap pemrograman. Pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu, pada fase ini juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum.
- **Pengujian.** Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.
- **Pendukung atau pemeliharaan.** Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

Gambar Waterfall :



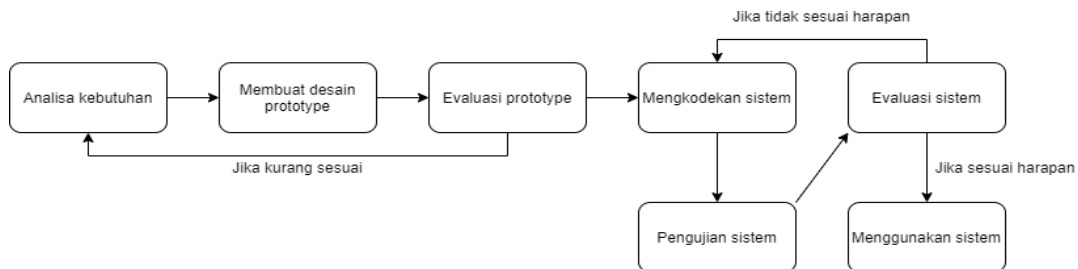
2. **Metode SDLC selanjutnya adalah prototype.** Metode **prototype** adalah metode yang memungkinkan pengguna atau user memiliki gambaran awal tentang perangkat lunak yang akan dikembangkan, serta pengguna dapat melakukan pengujian di awal sebelum perangkat lunak dirilis.

Metode ini bertujuan untuk mengembangkan model menjadi perangkat lunak yang final. Artinya sistem akan dikembangkan lebih cepat dan biaya yang dikeluarkan lebih rendah. Metode prototype ini memiliki tahap-tahap yang harus dilakukan dalam pengembangan perangkat lunak.

Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode prototype.

- **Analisa kebutuhan**
Pada tahap ini pengembang melakukan identifikasi perangkat lunak dan semua kebutuhan sistem yang akan dibuat.
- **Membuat prototype**
Membuat rancangan sementara yang berfokus pada alur program kepada pengguna.
- **Evaluasi prototype**
Evaluasi dilakukan untuk mengetahui apakah model prototype sudah sesuai dengan harapan.
- **Mengkodekan sistem**
Jika prototype disetujui maka akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.
- **Pengujian sistem**
Setelah perangkat lunak sudah siap, perangkat lunak harus melewati pengujian. Pengujian ini biasanya dilakukan dengan White Box Testing, Black Box Testing, dan lain-lain.
- **Evaluasi sistem**
Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.
- **Menggunakan sistem**
Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

Gambar Metode prototipe :



Pemilihan metode Siklus Hidup Pengembangan Perangkat Lunak (SDLC) untuk pembuatan sistem perpustakaan akan tergantung pada berbagai faktor, termasuk kompleksitas proyek, kebutuhan bisnis, dan sumber daya yang tersedia. Beberapa metode SDLC yang mungkin sesuai untuk proyek pembuatan sistem perpustakaan adalah:

- 1) Model Air Terjun (Waterfall): Jika kebutuhan sistem perpustakaan Anda telah cukup stabil dan tidak diharapkan banyak perubahan selama pengembangan, model Waterfall bisa efektif. Model ini mengikuti langkah-langkah yang terstruktur, mulai dari analisis hingga implementasi, dengan fokus pada dokumentasi yang kuat.
- 2) Model Incremental atau Iteratif: Proyek pembuatan sistem perpustakaan dapat mengambil pendekatan iteratif atau incremental jika ada kemungkinan kebutuhan yang berubah atau kompleksitas proyek yang tinggi. Pendekatan ini memungkinkan untuk perubahan perlahan dan evaluasi berkala, memungkinkan adaptasi selama pengembangan.
- 3) Model Spiral: Jika proyek memiliki aspek risiko yang signifikan, seperti integrasi dengan sistem lain atau keamanan yang tinggi, model spiral bisa cocok. Ini adalah pendekatan berulang yang fokus pada mitigasi risiko dan evaluasi secara berkala.
- 4) Metode Agile (seperti Scrum atau Kanban): Jika ada perubahan kebutuhan yang diperkirakan, dan jika ingin memastikan keterlibatan pengguna akhir dalam pengembangan, metode Agile bisa menjadi pilihan. Ini mengizinkan fleksibilitas tinggi dan memprioritaskan pengembangan berdasarkan kebutuhan bisnis yang paling penting.

Penting untuk melibatkan tim pengembangan dan pemangku kepentingan dalam pemilihan metode SDLC yang tepat. Selain itu, sebaiknya evaluasi komprehensif tentang kebutuhan, risiko, dan tujuan proyek sebelum memutuskan metode pengembangan yang paling sesuai untuk sistem perpustakaan Anda.

PEMBAHASAN 4 :

1. Latar belakang masalah dalam penelitian

Latar belakang masalah memuat penjelasan mengenai alasan-alasan masalah yang dikemukakan dalam penelitian yang dianggap menarik, penting dan perlu diteliti. Kedudukan masalah yang diteliti diuraikan juga dalam lingkup permasalahan yang lebih luas.

2. Rumusan masalah

Rumusan masalah merupakan permasalahan penelitian yang dijelaskan dalam bentuk pertanyaan dengan intensi untuk dijawab melalui proses penelitian yang akan dilakukan. Rumusan masalah boleh dibilang sebagai kunci atau inti dari proses penelitian.

3. Tujuan penelitian

Tujuan penelitian adalah ungkapan “mengapa” penelitian itu dilakukan. Tujuan dari suatu penelitian dapat untuk mengidentifikasi atau menggambarkan suatu konsep atau untuk menjelaskan atau memprediksi suatu situasi atau solusi untuk suatu situasi yang mengindikasikan jenis studi yang akan dilakukan.

Contoh :

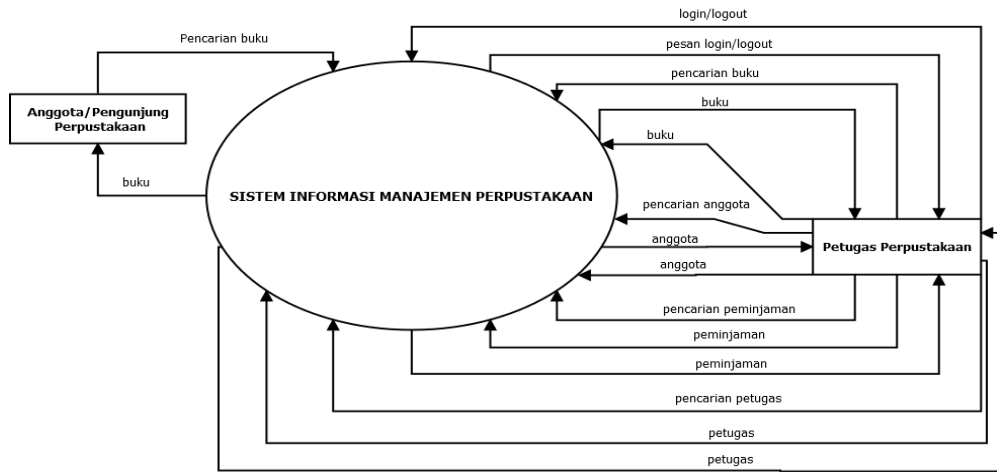
Latar Belakang Masalah:

Perpustakaan di perguruan tinggi memiliki peran kunci dalam mendukung pendidikan dan penelitian. Dalam konteks Universitas T Ut, perpustakaan adalah sumber informasi utama bagi mahasiswa, dosen, dan peneliti. Dalam beberapa tahun terakhir, perpustakaan telah mengalami perubahan besar dengan munculnya teknologi informasi. Namun, untuk terus efektif dalam menyediakan sumber daya dan layanan kepada pengguna, perlu memahami perubahan dan tantangan yang dihadapi sistem informasi perpustakaan.

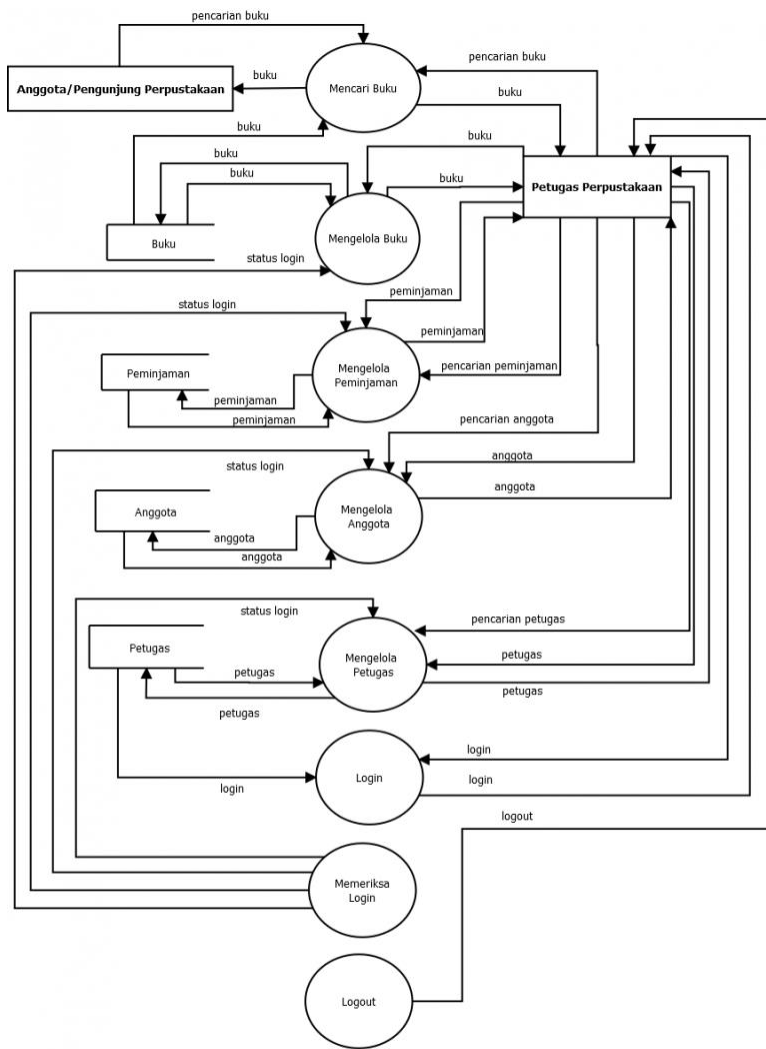
Rumusan Masalah: Bagaimana perubahan dalam teknologi informasi memengaruhi efektivitas sistem informasi perpustakaan di Universitas T Ut dan apa tantangan utama yang dihadapi dalam menghadapi perubahan tersebut?

Tujuan Penelitian: Tujuan penelitian ini adalah untuk:

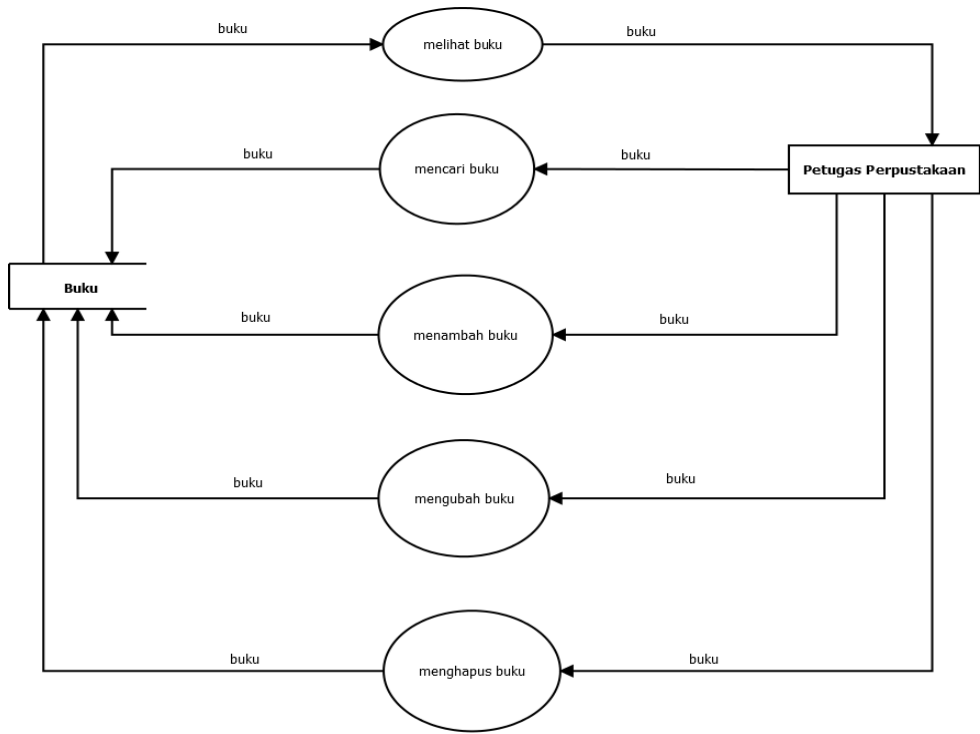
- 1) Menganalisis dampak perubahan teknologi informasi terhadap sistem informasi perpustakaan di Universitas T Ut.
- 2) Mengidentifikasi tantangan utama yang dihadapi sistem informasi perpustakaan di menghadapi perkembangan teknologi informasi.
- 3) Menyusun rekomendasi untuk meningkatkan efektivitas sistem informasi perpustakaan dan mengatasi tantangan yang diidentifikasi.



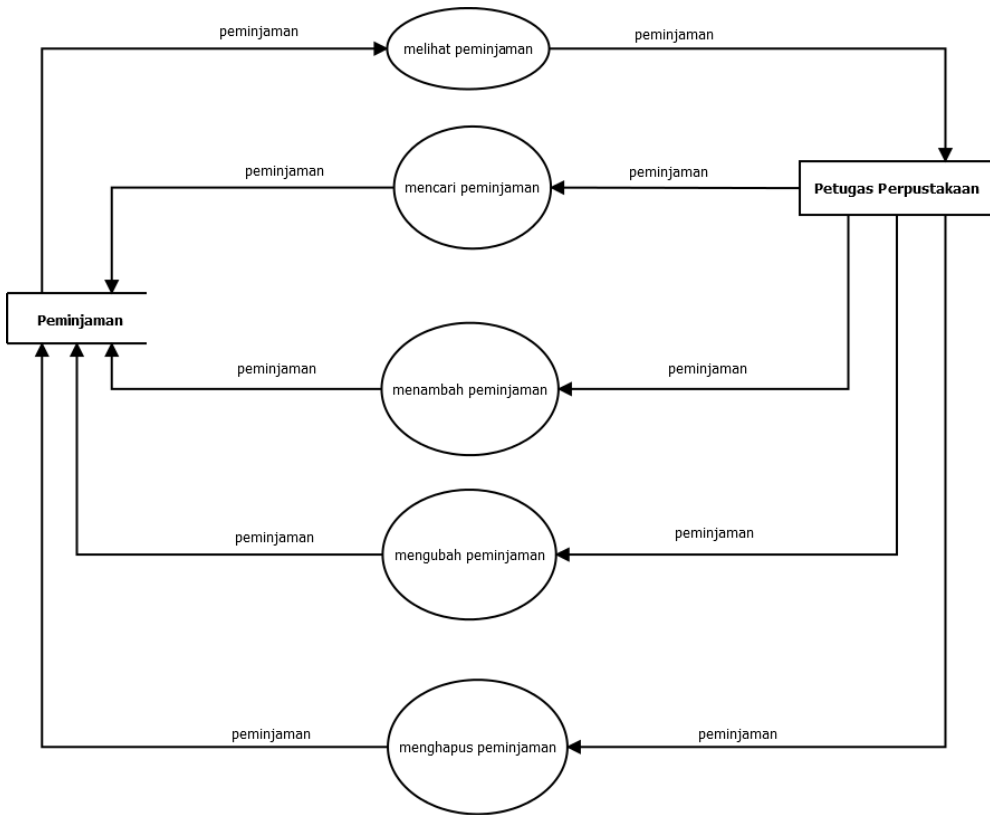
Gambar 1 : DFD level 0/Diagram Konteks



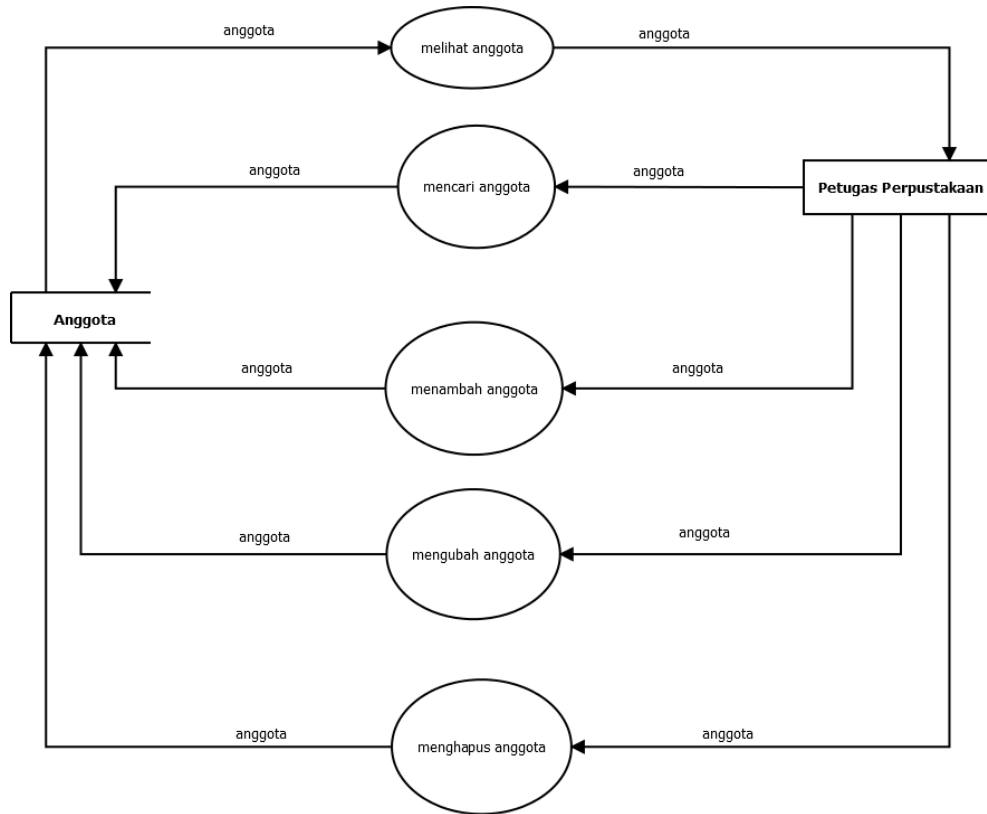
Gambar 2 : DFD level 1 Sistem informasi Perpustakaan



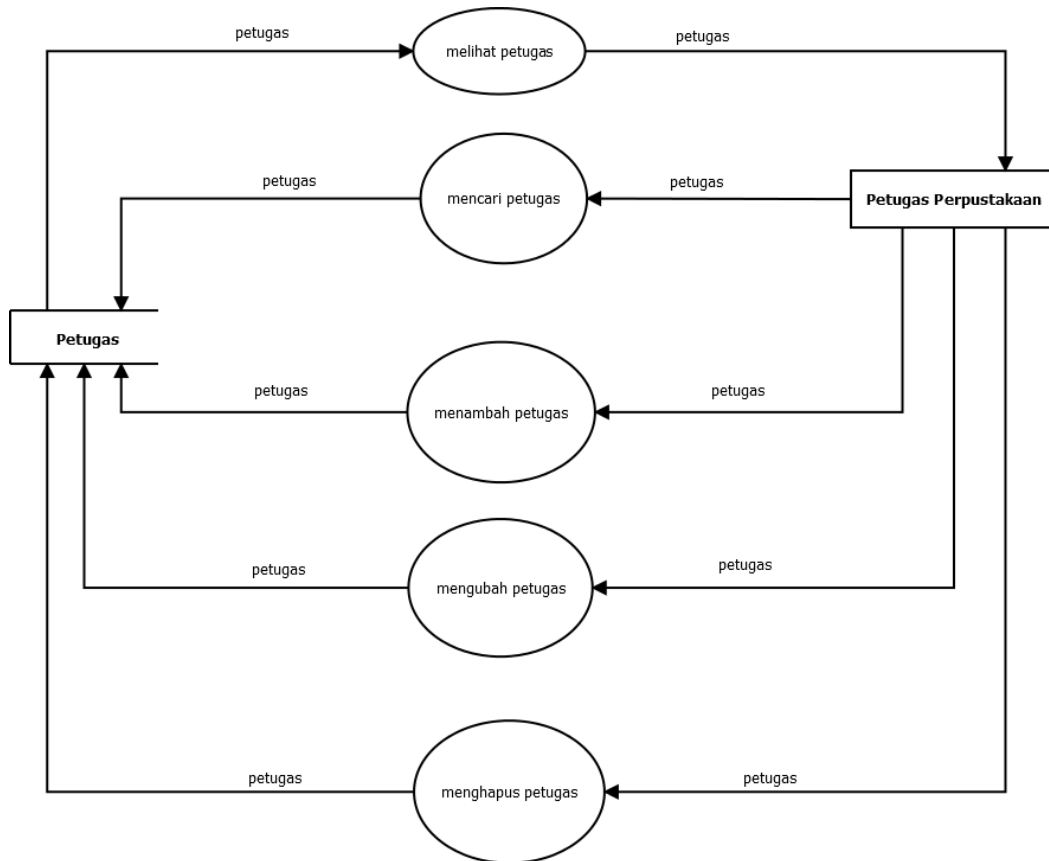
Gambar 3 : DFD level 2 Mengelola Buku



Gambar 4 : DFD level 2 Mengelola Peminjaman



Gambar 5 : DFD level 2 Mengelola Anggota



Gambar 6 : DFD level 2 Mengelola Petugas