

## ESSAY

1. **Model Relasional** sendiri adalah suatu model basis data yang menggunakan table dua dimensi, yang terdiri atas baris dan kolom untuk menggambarkan sebuah berkas data.

**Model Relasional** merupakan model data yang paling banyak digunakan saat ini **karena model relasional menghasilkan representasi data yang disusun dalam baris dan kolom secara konsisten dan logis**, yang membuat data tersebut bisa diakses dan dihubungkan ke baris dan table lainnya dengan membagikan **common field** atau yang kita kenal dengan **PRIMARY KEY** dan **FOREIGN KEY**.

Alasan lainnya mengapa model relasional sering dipakai adalah:

- a) Memungkinkan programmer untuk mengakses data melalui relationship antar files. Relasi baru antar file bisa ditambahkan dengan mudah
- b) Model relasional lebih fleksibel dan memberikan relasi yang natural antar data. (Hal ini tidak seperti model hirarki yang hanya mengizinkan relasi **one-to-one** dan **one-to-many**)
- c) Model relasional tidak memerlukan database untuk diurutkan ulang ketika sebuah data baru itu ditambahkan
- d) Berkurangnya kompleksitas karena perubahan bisa dilakukan terhadap skema tanpa harus mempengaruhi aksesibilitas data
- e) Model relasional **mendukung kueri ad-hoc** dan **reports**.

Ketika menggunakan model relasional, kita bisa dengan mudah melakukan organisasi data, bisa dengan mudah melakukan pembuatan data, mudah untuk menarik data, data integrity yang sudah terbangun didalamnya, fleksibilitas kueri (seperti yang sudah disebutkan diatas).

Dengan model relasional, seseorang bisa dengan mudah melakukan **Data Query Operation** seperti (yang sudah saya pelajari dari matakuliah 'Database Systems'): **CRUD (Create-Read-Update-Delete)**, dan **juga Insert**.

Inilah yang menjadi factor-faktor pada umumnya mengapa model relasional ini lebih dipakai daripada model network dan model hirarki.

2. Komponen navigasi dari sebuah interface memungkinkan user untuk memasukkan command (perintah) untuk menjelajahi sebuah system dan menjalankan sebuah fungsi dalam system serta memperoleh informasi yang ditunjukkan system. Komponen navigasi juga memberikan pesan kepada user tentang apakah tindakan yang dilakukan oleh user dalam system itu berhasil atau tidak.

Tujuan dari system navigasi ini adalah untuk memudahkan user menggunakan dan memahami system dengan lebih baik.


Salah satu system navigasi yang paling umum dipakai dalam aplikasi perangkat lunak saat ini adalah **Menu**. Konsep **Menu** sendiri adalah: menyajikan pengguna dengan daftar pilihan, yang masing-masing dapat dipilih oleh user.

**Mengapa menggunakan Menu?** Karena menu lebih mudah dipelajari karena perintah-perintah itu sudah tersedia ditampilkan kepada pengguna secara tertata dalam berbagai tipe menu untuk fungsi yang berbeda-beda.

Jenis menu yang dapat kita terapkan pada aplikasi adalah: ***menu bars, drop-down menu, pop-up menu, tab menu, tool bar, dan image maps.***

(Alan Dennis, Barbara Haley Wixom: Systems Analysis & Design, An Object-Oriented Approach with UML. 5.)

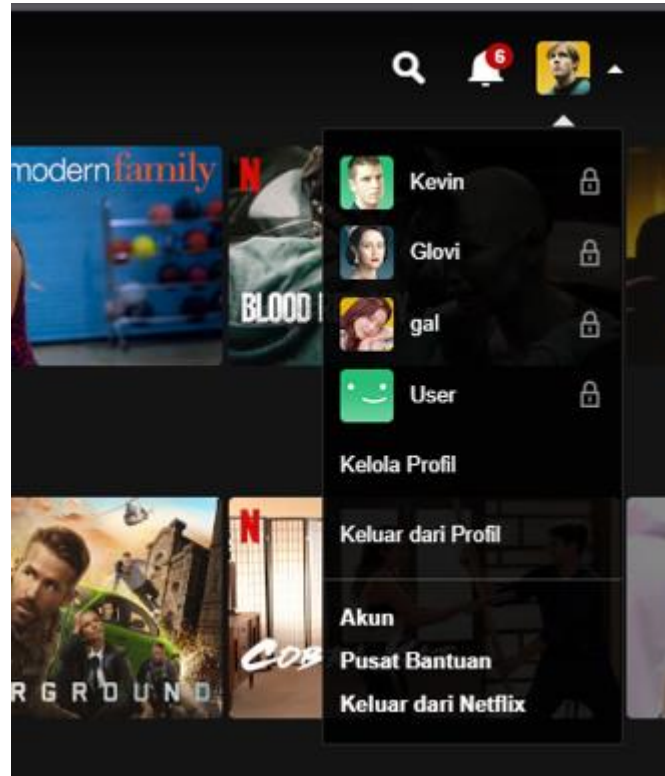
Berikut contoh-contoh menu beserta penjelasannya:

Jenis Menu	Penjelasan
Menu Bars	<p data-bbox="445 274 1759 347">Sebuah area yang menampilkan beberapa main menu. Penampilan menu bar ini bisa secara Horizontal, bisa juga secara Vertikal.</p> <p data-bbox="445 380 961 412">Contoh menu bar secara Horizontal:</p> <div data-bbox="445 485 1740 828">  </div> <p data-bbox="445 834 1474 867"><b>CHARACTERS – COMICS – MOVIES – TV SHOWS – GAMES - NEWS</b></p> <p data-bbox="445 907 924 980">Dari website:  <a data-bbox="445 948 924 980" href="https://www.marvel.com/unlimited">https://www.marvel.com/unlimited</a> </p>

Menu bar:  
**VIDEOS**  
–

Drop-  
down  
menu

Menu yang menampilkan sebuah list atau daftar opsi-opsi yang dapat dipilih user  
Contoh menu:



(Ketika diklik segitiga putih kecil, maka akan menampilkan pilihan profil lain (Kevin, Glovi, dst) lalu menampilkan juga pilihan 'Kelola Profil' dan 'Keluar dari Profil'.

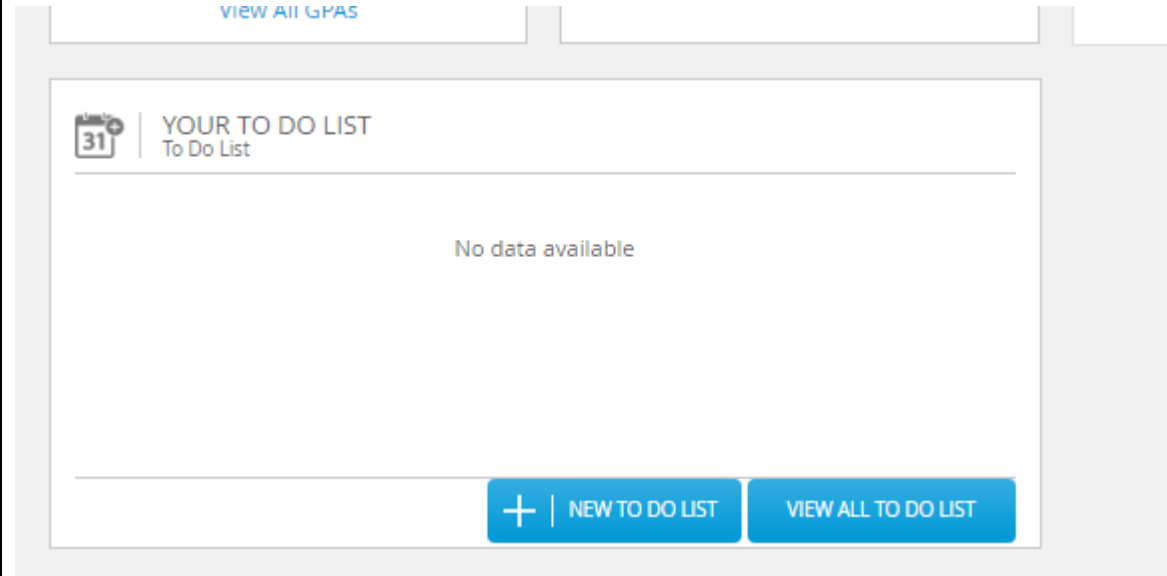
Dari website: <https://www.netflix.com/>

Pop-up  
menu

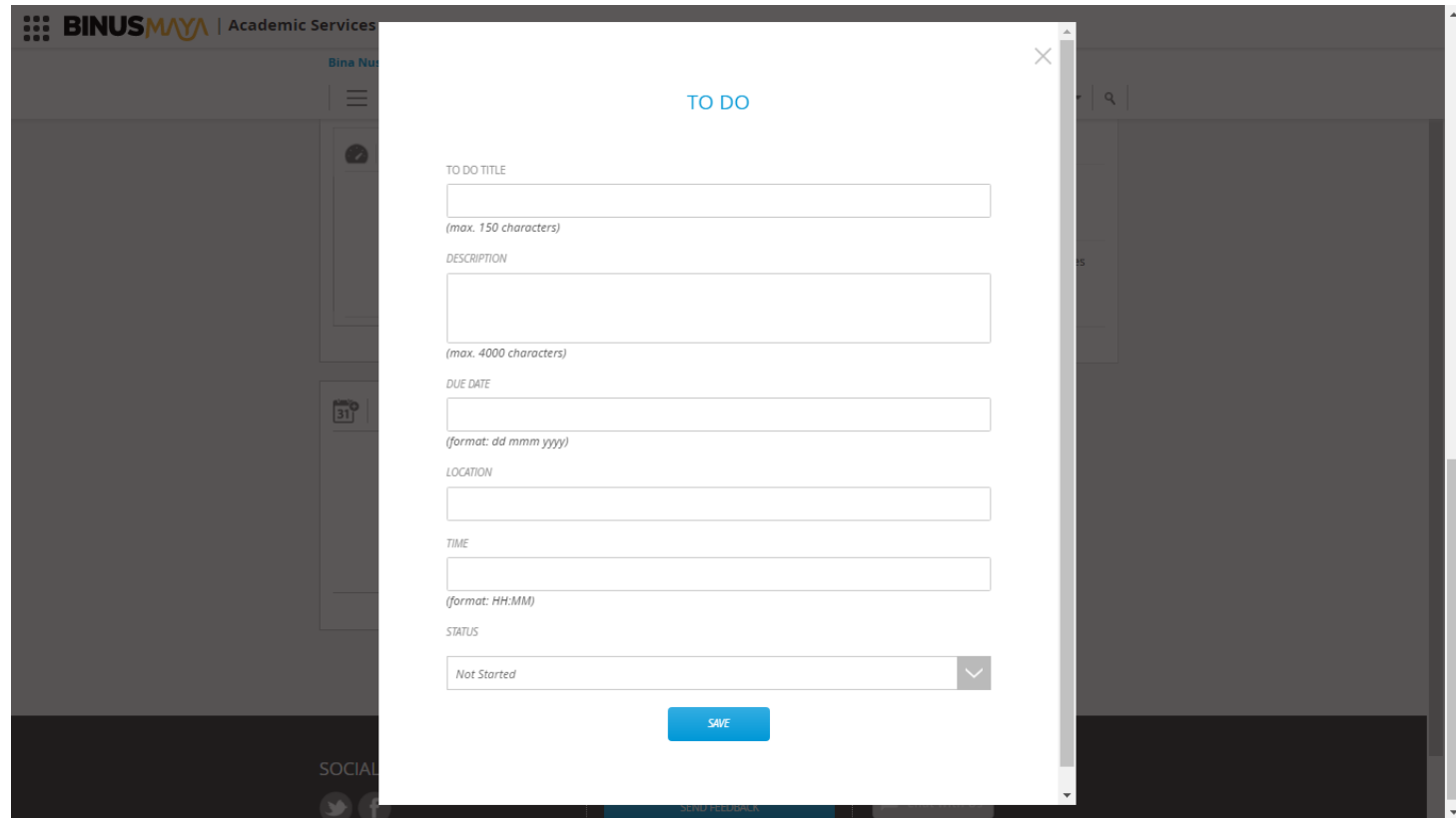
Pop-up menu adalah menu yang tersembunyi dan tidak muncul sampai kita klik kanan terhadap suatu tombol.

Contoh menu:

Misalkan saya mengklik kanan tombol 'New to Do List':



Maka akan muncul popup menu 'TO DO':



The screenshot displays a web interface for BINUSMAYA Academic Services. A 'TO DO' popup form is centered on the screen, overlaying the background. The form includes the following fields and options:

- TO DO TITLE**: A text input field with a placeholder '(max. 150 characters)'.
- DESCRIPTION**: A text input field with a placeholder '(max. 4000 characters)'.
- DUE DATE**: A date input field with a placeholder '(format: dd mmm yyyy)'.
- LOCATION**: A text input field.
- TIME**: A time input field with a placeholder '(format: HH:MM)'.
- STATUS**: A dropdown menu currently showing 'Not Started'.
- SAVE**: A blue button at the bottom of the form.

The background website shows the BINUSMAYA logo, 'Academic Services' text, and a sidebar with icons for navigation. The bottom of the page features social media links and a footer.

Dari website:

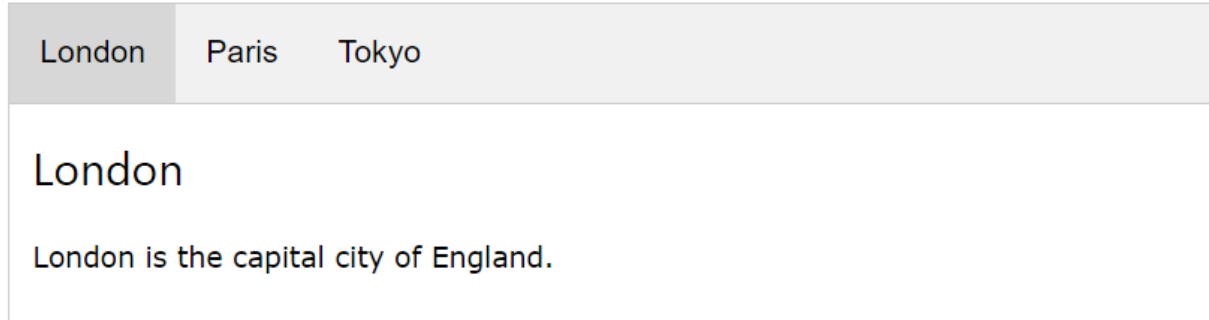
<https://binusmaya.binus.ac.id/>

Tab  
menu

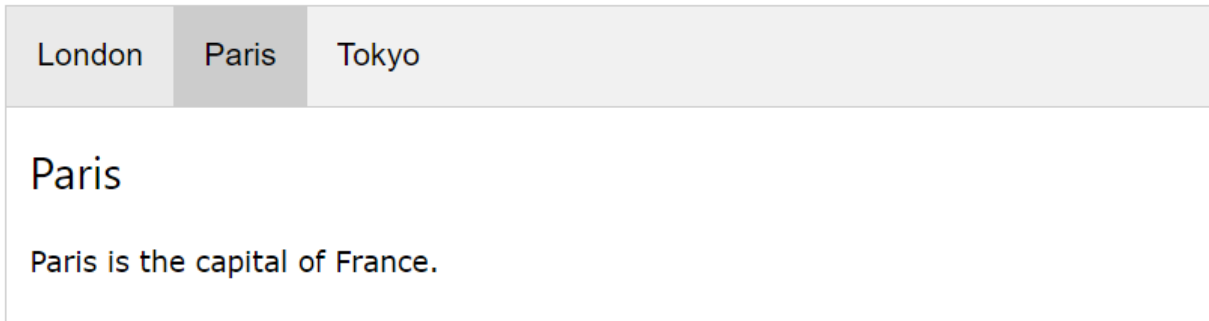
Yakni Tabs (menu untuk menampilkan halaman-halaman berbeda untuk sebuah konten dalam aplikasi)

Contoh menu:

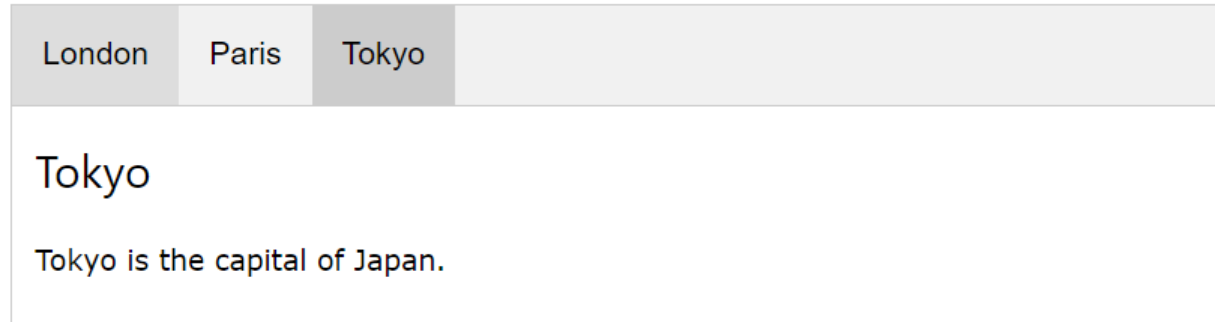
Misal pertama kita berada di tab 'London' untuk halaman London:



Kemudian kita klik tab 'Paris', kita jadinya berada di halaman 'Paris':



Begitu juga ketika kita klik tab Tokyo, maka akan menampilkan halaman Tokyu:



Dari website:

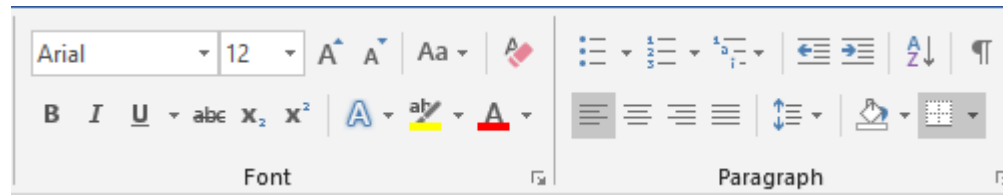
[https://www.w3schools.com/howto/howto\\_js\\_tabs.asp](https://www.w3schools.com/howto/howto_js_tabs.asp)

Tool bar

Tool bar dengan menu bar hampir sama. Yang membedakan mereka adalah tool bar mengandung gambar atau icon, sedangkan menu bar hanya mengandung kata.

Contoh menu:

Seperti ada icon huruf kapital B dengan penulisan tebal, menandakan 'Bold' untuk mempertebal tulisan.



Dari:

Interface Microsoft Word 2016



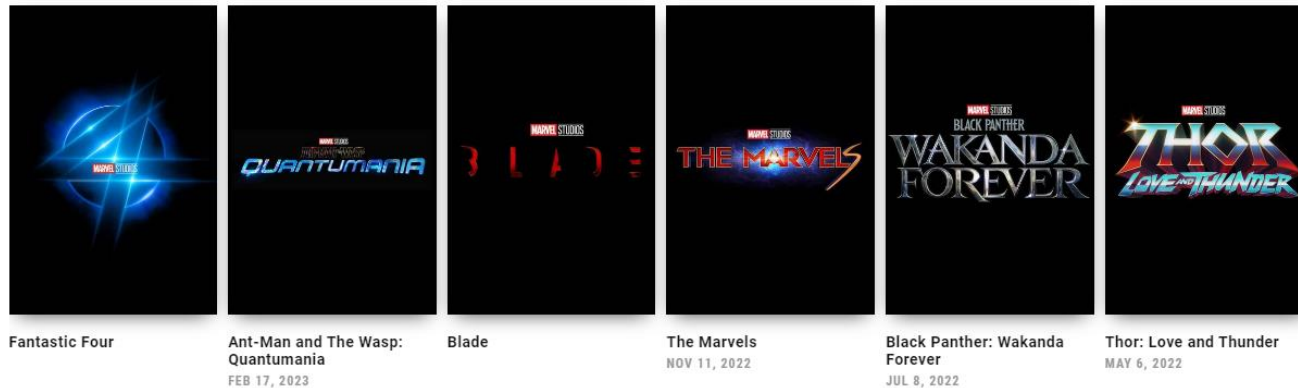
Image  
Maps

Dalam HTML, image map itu seperti Namanya 'Image' menggunakan sebuah gambar, yang dibuat untuk menuju ke situs lainnya atau ke tempat yang berbeda.

Contoh menu:

Dari website <https://www.marvel.com/unlimited>, ditampilkan poster-poster film. Ketika kita mengklik salah satu poster film tersebut (misalkan kita mengklik poster 'Thor':

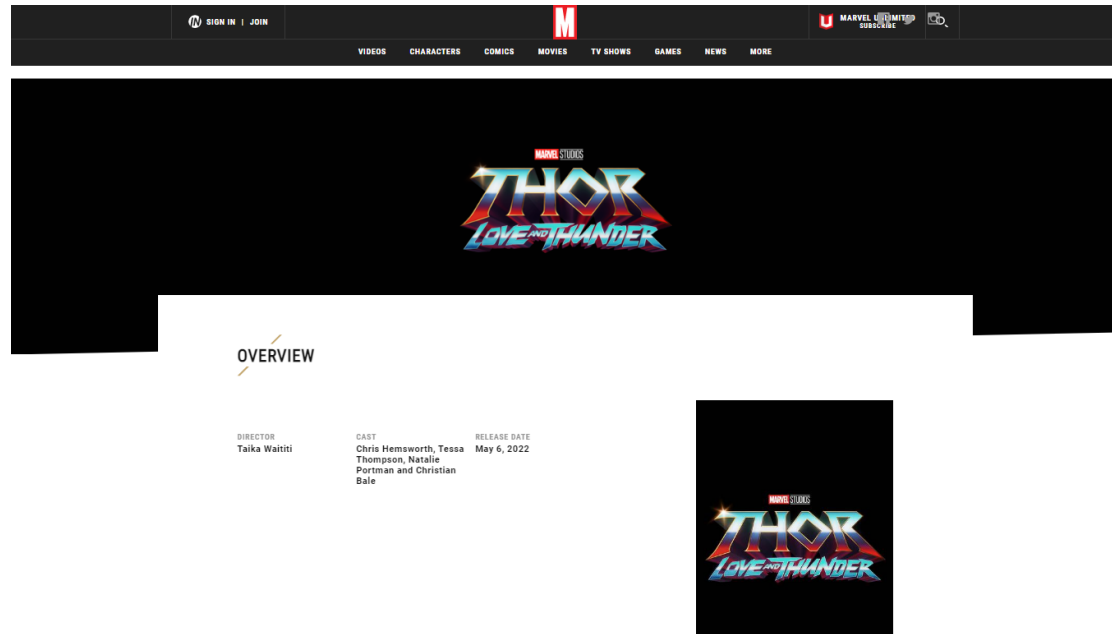
### MARVEL MOVIES



Website address mula-mula:

<https://www.marvel.com/movies>

Maka, kita akan diarahkan ke area lainnya, yakni:



Website setelah mengklik image:

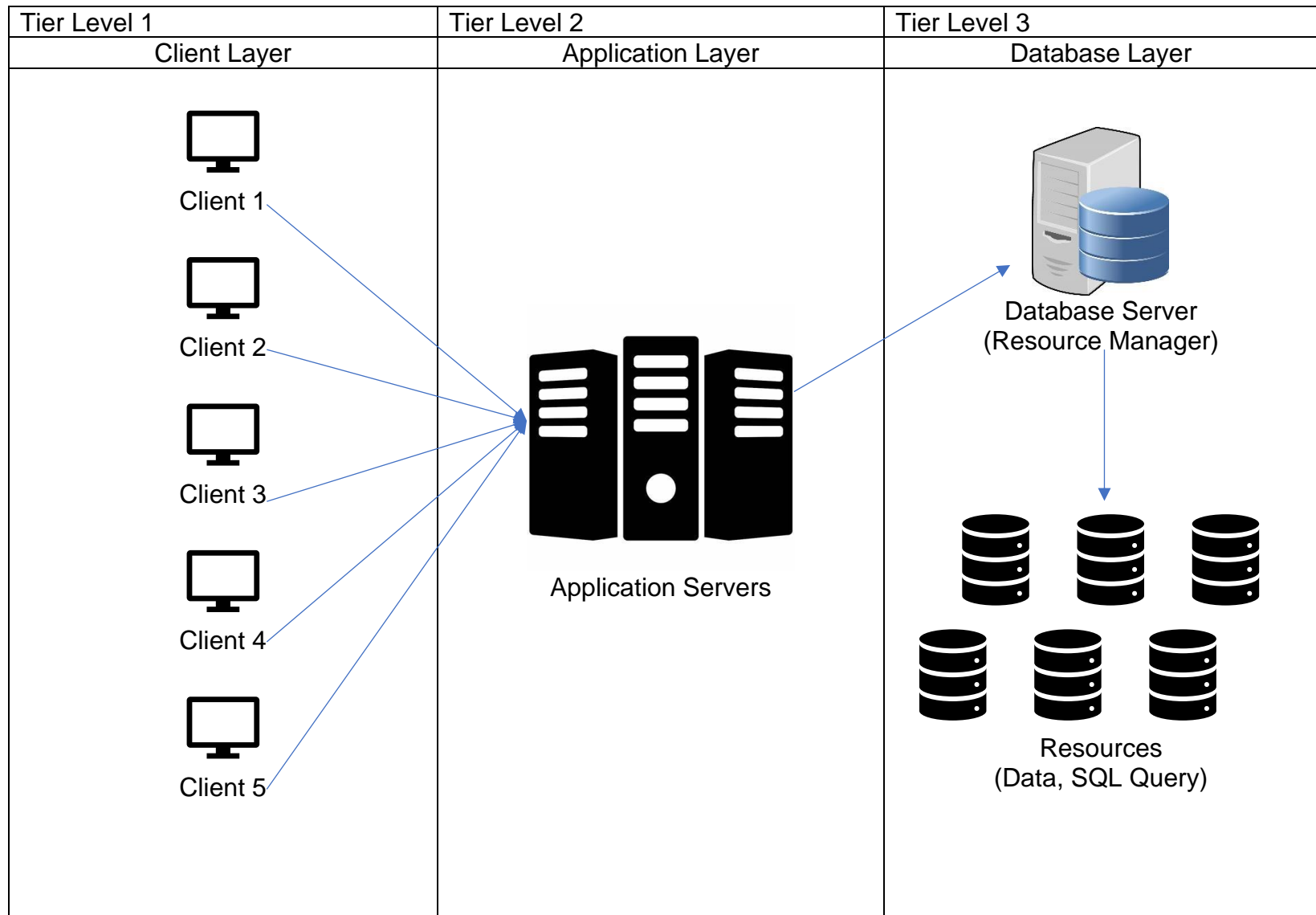
<https://www.marvel.com/movies/thor-love-and-thunder>

3. Semua software system bisa dibagi menjadi empat functions dasar.

Basic Functions	Description
Data Storage	Kebanyakan informasi system itu memerlukan data untuk disimpan dan ditarik (diterima), baik itu file berukuran kecil maupun database berukuran besar.  <b>“Menyimpan data yang dihasilkan oleh program (file, record)”</b>
Data Access Logic	Sebuah proses itu memerlukan <b>akses data</b> , biasanya diartikan sebagai kueri database dalam SQL.  <b>“Mengakses Data dari resources”</b>
Application Logic	Logic yang didokumentasikan pada Data Flow Diagram, Use Case, dan <i>functional Requirement</i> .  <b>“Business Logic’, yakni proses yang dijalankan dalam aplikasi”</b>
Presentation Logic	Sebuah tampilan informasi kepada user dan penerimaan perintah (input) dari user (user interface).  <b>“Menampilkan informasi-informasi pada layar computer (hardware) pengguna”</b>

Dalam soal ini, saya akan memilih untuk mencontohkan lewat Three Tier Client Server Architectures

Contoh penerapannya:



Penerapan Basic Function		
Client Layer	Application Layer	Database Layer
Responsible Basic Function:  <b>PRESENTATION LOGIC</b>	Responsible Basic Function:  <b>APPLICATION LOGIC</b>	Responsible Basic Function:  <b>DATA ACCESS LOGIC DATA STORAGE</b>

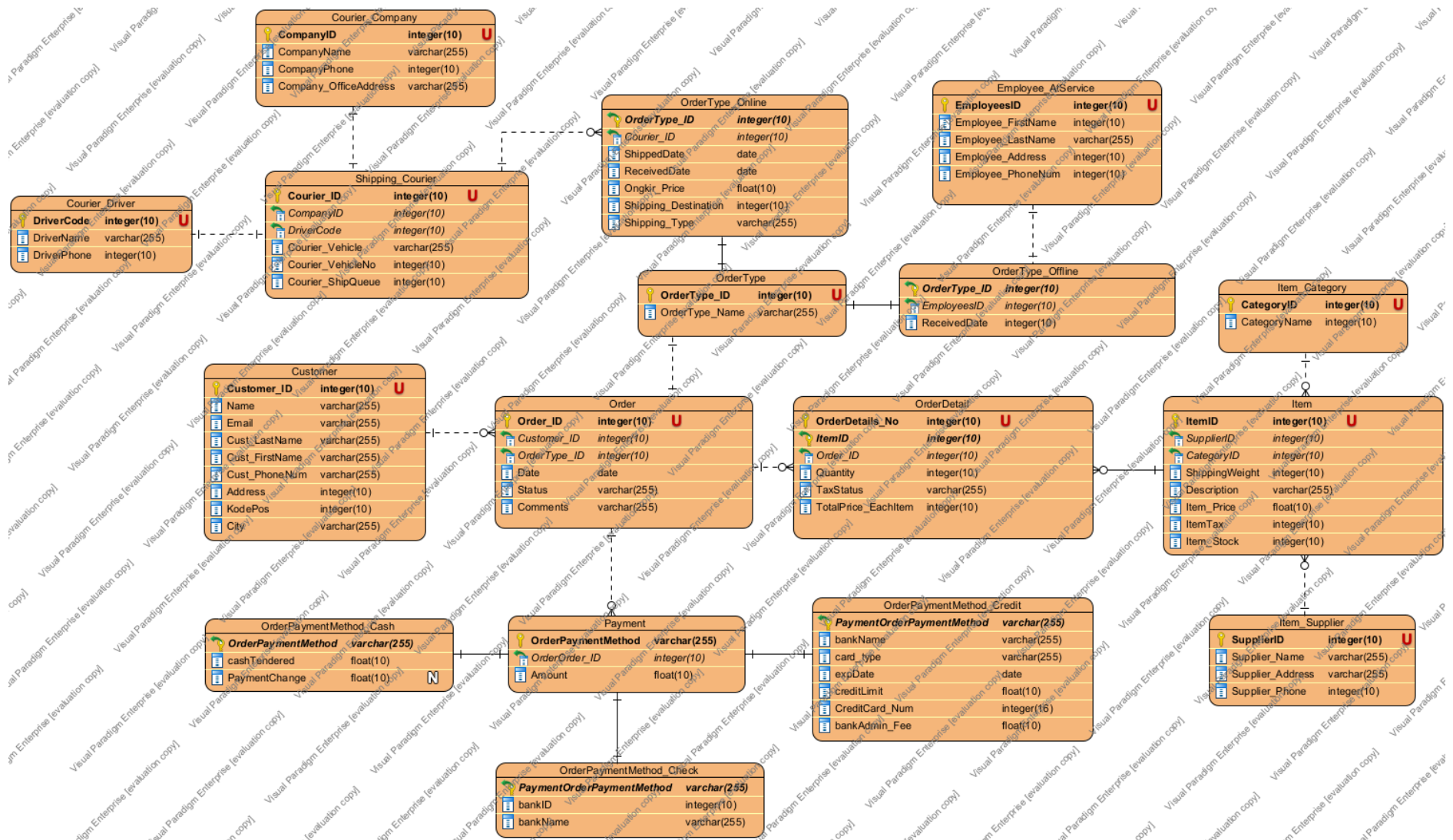
Penjelasan:

Pada Client Layer, software yang ada pada computer client bertugas sebagai “yang menampilkan informasi kepada user” (**Presentation Layer**)

Pada Application layer, application server bertugas sebagai: **multiple processing functions**, yakni yang mengkoordinasikan domain dan infrastruktur berdasarkan requirement aplikasi. (**application logic**)

Pada, database layer, **data diakses dari resources (SQL) dan ada operasi terhadap data tersebut**. Disini ia menjalankan “**Data Access Logic**”. Juga, dengan adanya penyimpanan data dari program, maka menjalankan fungsi “**Data Storage**” (Penyimpanan Data)

#### 4. Diagram Model Relasi (tersedia file .png) [Terdapat watermark ketika membuat di visual paradigm]:



Penjelasan untuk relasi yang dimiliki:

1. Satu customer dapat memiliki banyak order, (one-to-many)
2. Satu order dapat memiliki banyak details seperti item-item yang dibeli beserta quantity yang dibeli (one-to-many)
3. Untuk semua order detail, informasi tentang item hanya ada satu-satu untuk masing-masing item (many-to-one)
4. Banyak item dapat memiliki satu kategori yang sama (one-to-many)
5. Banyak item dapat di supply (diproduksi) oleh satu vendor (many-to-one)
6. Satu order hanya bisa memiliki satu metode pembayaran, entah itu Cash, Check, Atau Kredit (one-to-one)
7. Satu order hanya bisa memiliki satu tipe orderan, entah itu secara online atau offline.
8. Untuk tipe order online, banyak order bisa dilakukan oleh satu kurir driver (one-to-many)
9. Seorang driver hanya bisa memiliki satu kode driver dalam kurir tersebut (one-to-one)
10. Seorang driver hanya bisa terdaftar pada satu perusahaan kurir (Courier\_Company) – (one-to-one)
11. Selain OrderType\_Online, ada OrderType\_Offline, dimana disini menyimpan informasi tentang siapa pegawai yang melayani orderan tersebut
12. Banyak order dapat dilakukan oleh satu employee (Employee\_AtService) – (one-to-many)

Berikut beberapa asumsi yang saya tambahkan:

- System order yang dilakukan bisa online dan offline.

Mengapa bisa online? Karena pada kelas Item, terdapat **ShippingWeight**. Ada aktivitas pengiriman 'Shipping'.

Lalu kenapa ada Offline? karena **ada berbagai macam metode pembayaran**, salah satunya dengan **Cash**. Dimana ini hanya bisa dilakukan ketika Offline saja.

- Karena ada order secara online, maka diperlukan kurir yang mengantar order tersebut.
- Karena ada order secara offline, maka diperlukan pegawai (Employee\_AtService) yang melayani order tersebut

- Juga ada atribut yang saya tambahkan untuk semua kelas agar proses pengaksesan data dapat dilakukan secara lebih efektif dan juga untuk menambah kejelasan informasi kelas.

Karena asumsi-asumsi baru tersebut, maka entity-entity baru yang saya tambahkan:

- OrderType  
(tipe order yang dilakukan)
- OrderType\_Online  
(untuk tipe order online)
- OrderType\_Offline  
(untuk tipe order offline)
- Shipping\_Courier  
(untuk kurir yang melakukan pengiriman order online)
- Courier\_Driver  
(informasi tentang driver yang mengirimkan)
- Courier\_Company  
(informasi tentang perusahaan yang menaungi driver)
- Employee\_AtService  
(informasi tentang pegawai yang melayani order offline yang dilakukan oleh customer)
- Item\_Category  
(Dalam mengakses secara online, customer perlu mencari item, dan agar dimudahkan dalam proses pencarian, tiap-tiap item diberi kode kategori)
- Item\_Supplier  
(Untuk setiap item ditambahkan informasi tentang asal item tersebut)



5. Deployment Diagram untuk Library Management System [Terdapat watermark ketika membuat di VisualParadigm]:

