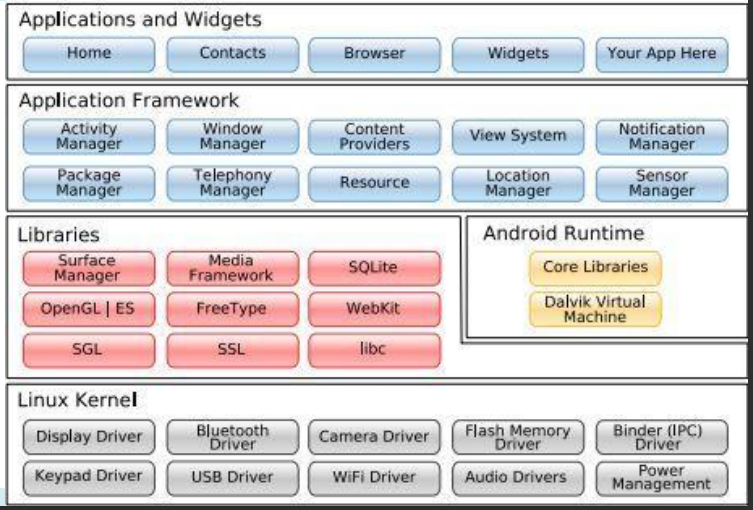
ARSITEKTUR SISTEM TERDISTRIBUSI

Arsitektur 🡪 suatu rancangan untuk penyususnan komponen-komponen suatu system, dimana rancangan tersebut mengidentifikasi komponen serta fungsi masing-masing komponen, konektifitas antar komponen serta pemetaan fungsionalitas komponen.



* Model Arsitektur Sistem Terdistribusi :

1. Arsitektur Logis (Software architecture)

Organisasi logika dari komponen-komponen perangkat lunak. Komponen yang dimaksud itu adalah unit modular berupa interface yang dapat diproses di sistem yang berbeda. Contohnya seperti RPC (remote prosedure call), message passing.

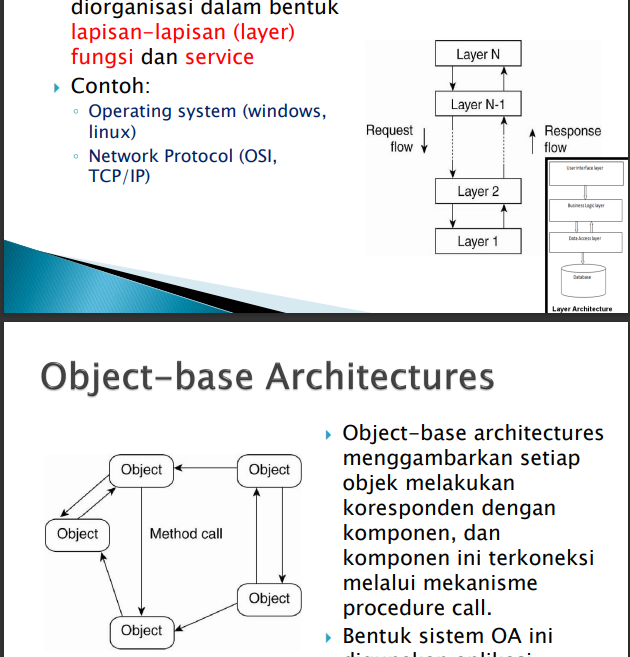
Jenis Model Arsitektur Logis (style):

* Layered architectures

Komponen-komponen diorganisasi dalam bentuk lapisan-lapisan (layer) fungsi dan service.

Contoh :

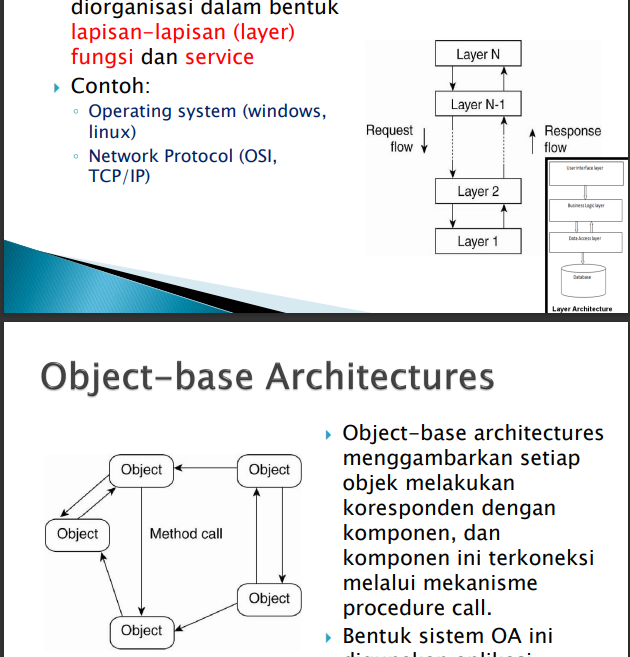
Operating system (windows, linux), Network Protocol (OSI, TCP/IP)



* Object-base architectures

Menggambarkan setiap objek melakukan koresponden dengan komponen, dan komponen ini terkoneksi melalui mekanisme procedure call. Bentuk sistem OA ini

digunakan aplikasi perangkat lunak dalam skala besar.

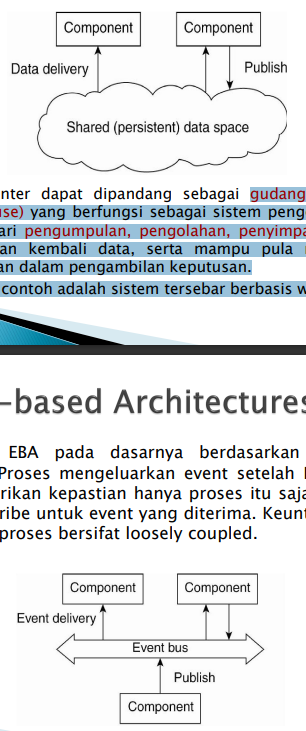


* Data-Center architectures

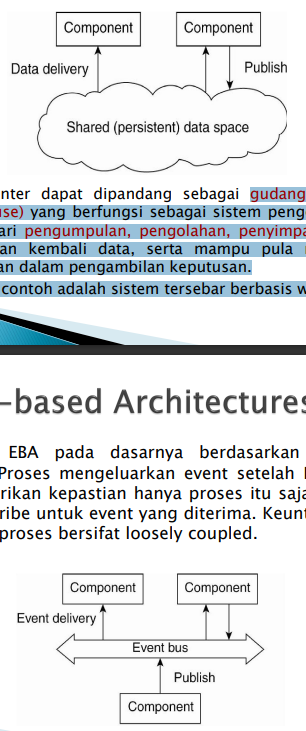
Gudang data (data warehouse) yang berfungsi sebagai sistem pengelolaan data mulai dari pengumpulan, pengolahan, penyimpanan hingga penemuan kembali data, serta mampu pula memberikan dukungan dalam pengambilan keputusan.

Contoh :

Sistem tersebar berbasis web



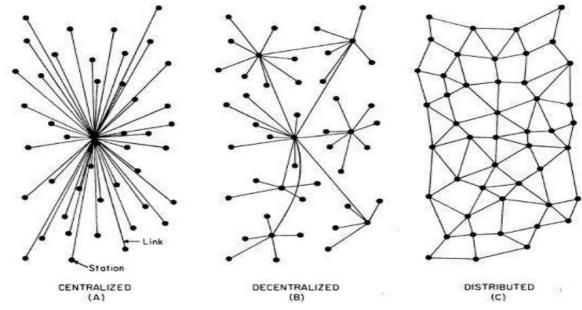
* Event-based architecturesProses mengeluarkan event setelah Middleware memberikan kepastian hanya proses itu saja yang bisa di subcribe untuk event yang diterima. Keuntungan EBA adalah proses bersifat loosely coupled.



1. Arsitektur Fisik (System architecture)

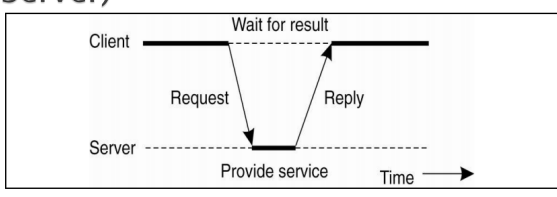
Peletakan komponen perangkat lunak pada mesin sesungguhnya.

* System Architecture



1. Centralized Architectures (Client-Server)

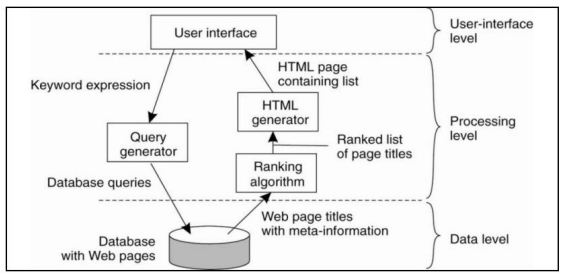
Client server unggul dalam kecepatan dan mendukung jaringan besar dan terdapat perilaku yang biasa disebut request-reply behavior. Namun memiliki kekurangan yaitu pada sisi setup yang cukup komplek, biaya tinggi dan membutuhkan sumber daya manusia yang handal untuk mengelola.



* Application Layering

Pada umumnya client server architecture ditujukan untuk keperluan user access ke database, maka dari itu layered architectural style dibagi menjadi:

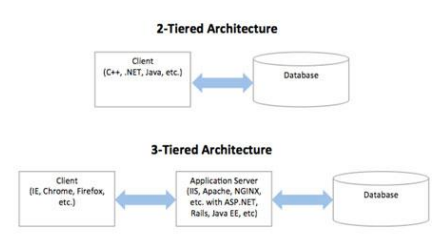
* user-interface level (display management)
* processing level (applications)
* data level (actual data that is being acted on)



* Multi-tiered Architectures
* 2 tier architecture

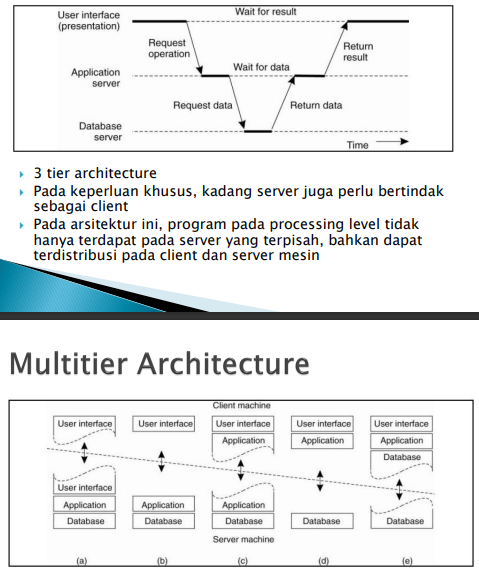
Pengorganisasian paling simple dimana terdiri atas 2 type mesin :

* Client yang berisi implementasi program pada user-interface level
* Server yang berisi implementasi program pada proses dan data level



* 3 tier architecture

Program pada processing level tidak hanya terdapat pada server yang terpisah, bahkan dapat terdistribusi pada client dan server mesin

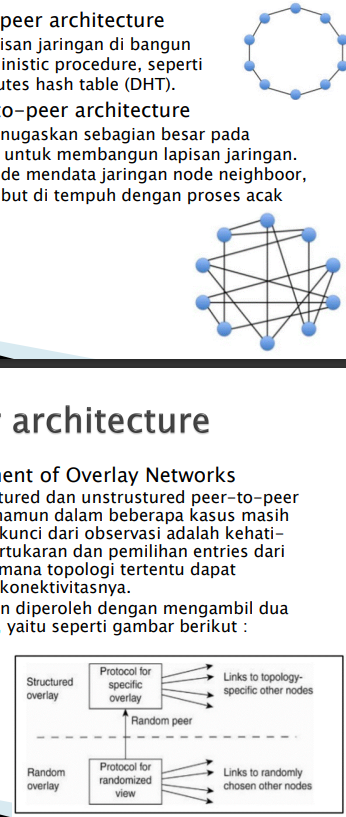


1. Decentralized Architectures

Disebut sebagai Horizontal distribution. secara fisik terpisah namun secara logis memiliki fungsi level yang sama (equivalent), dimana setiap mesin memproses bagiannya sendiri kemudian melakukan balancing terhadap hasil proses.

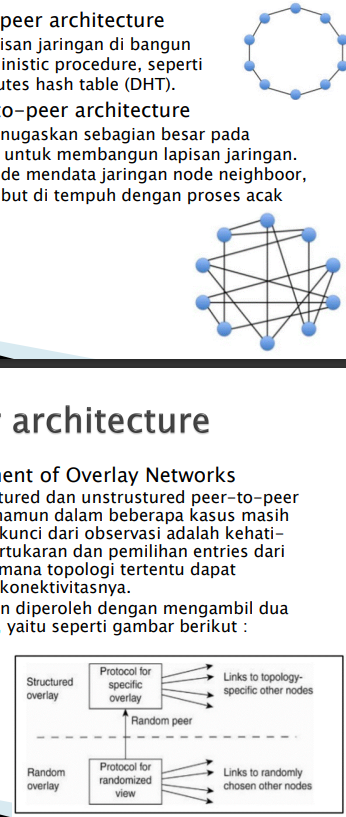
* Structured P2P (Peer-to-Peer) Architectures

lapisan jaringan di bangun menggunakan deterministic procedure, seperti menggunakan distributes hash table (DHT).



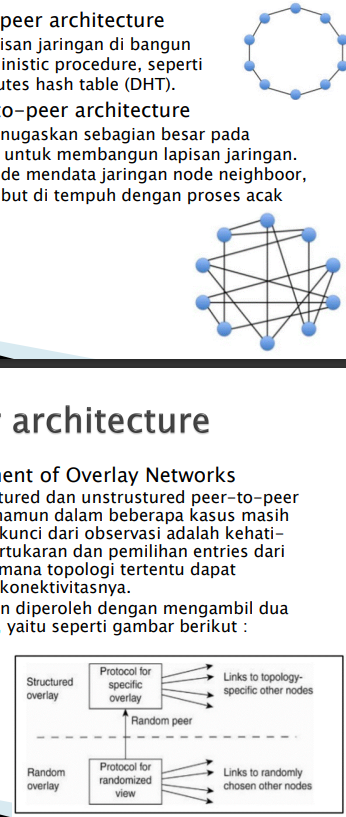
* Unstructured P2P Architectures

menugaskan sebagian besar pada algoritma secara acak untuk membangun lapisan jaringan. Pada intinya setiap node mendata jaringan node neighboor, tetapi data node tersebut di tempuh dengan proses acak sederhana.



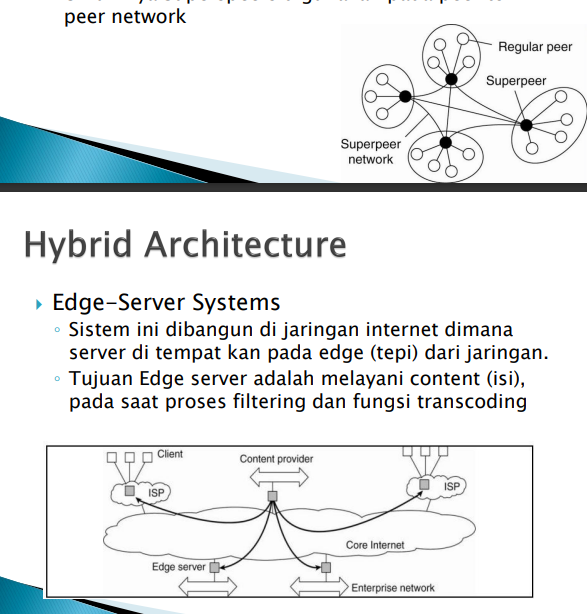
* Topology Management of Overlay Networks

Satu kunci dari observasi adalah kehatihatian dari proses pertukaran dan pemilihan entries dari pandangan parsial dimana topologi tertentu dapat dibangun dan dijaga konektivitasnya. Pendekatan TMOOD in diperoleh dengan mengambil dua pendekatan Layering, yaitu seperti gambar berikut :



* Superpeers

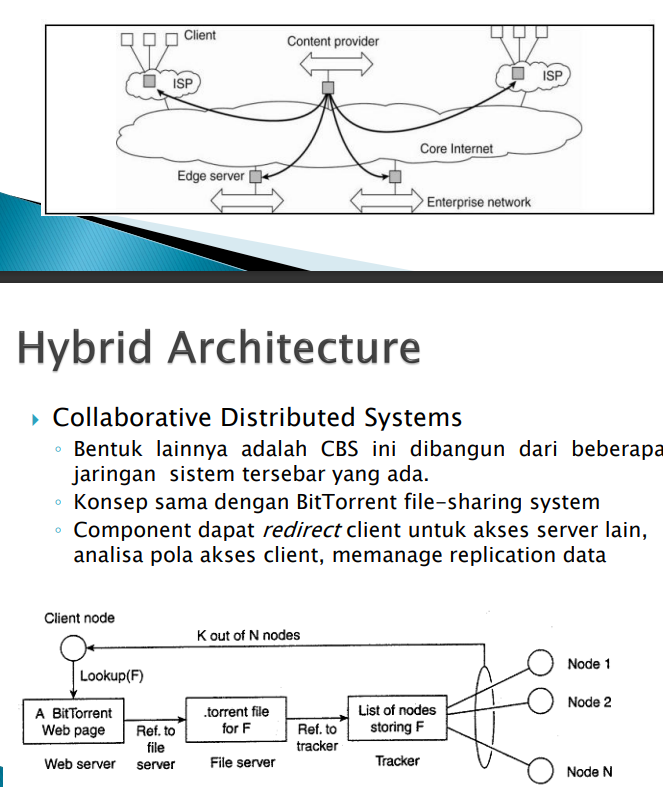
Teknik Superspeers biasanya menangani masalah yang disebabkan penempatan item data ketika jaringan berkembang dan juga dapat menangani masalah yg terkait dengan scalability, karena dapat mempertahankan konektifitas terhadap item data. Umumnya Superspeers digunakan pada peer to peer network.



1. Hybrid Architectures

* Edge-Server Systems

Sistem ini dibangun di jaringan internet dimana server di tempat kan pada edge (tepi) dari jaringan. Tujuan Edge server adalah melayani content (isi), pada saat proses filtering dan fungsi transcoding.



* Collaborative Distributed Systems

CBS ini dibangun dari beberapa jaringan sistem tersebar yang ada. Component dapat redirect client untuk akses server lain, analisa pola akses client, memanage replication data.

