

SISTEM TERDISTRIBUSI
“ARSITEKTUR SISTEM TERDISTRIBUSI”



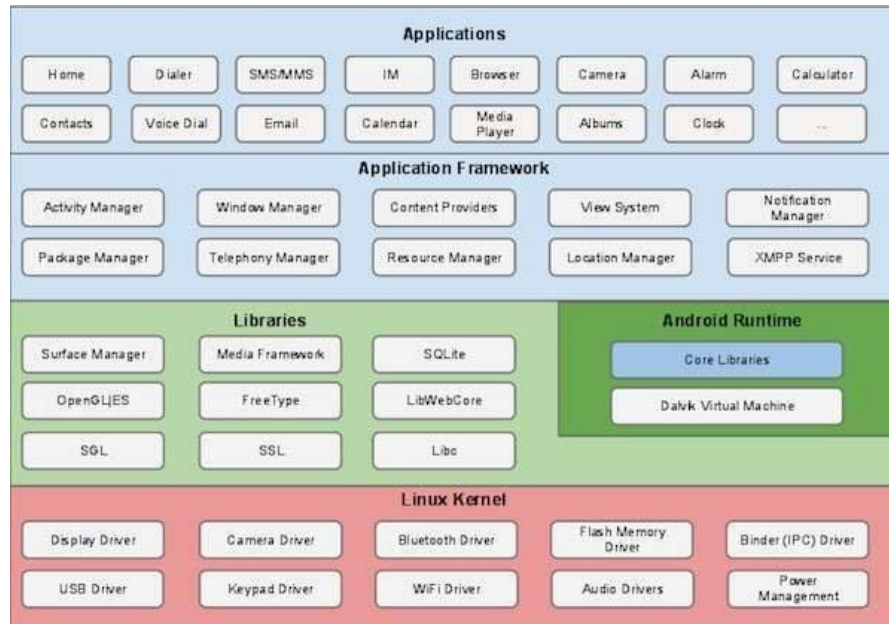
DISUSUN OLEH:
DINDA GATYA RABBANI
2111082012
TRPL 3B

DOSEN PENGAMPU :
ERVAN ASRI, S.Kom., M.Kom

SEMESTER V
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK
POLITEKNIK NEGERI PADANG

Arsitektur Sistem Terdistribusi

Suatu rancangan untuk penyusunan komponen-komponen suatu sistem, 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 berupa unit modular berupa interface yang dapat diproses di sistem yang berbeda

2) Arsitektur Fisik(System Architecture)

- Peletakan mesin
- Peletakan komponen perangkat lunak pada mesin sesungguhnya

Layered Architectures

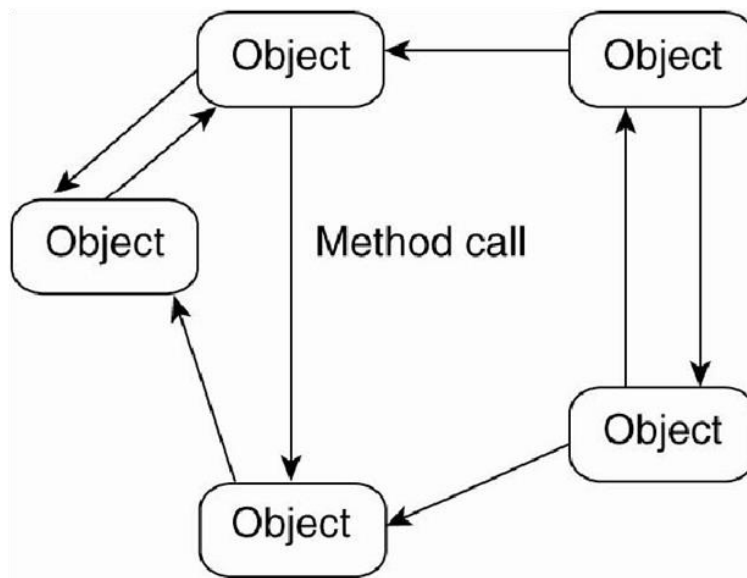
Komponen-komponen pada Layered architectures diorganisasi dalam bentuk lapisan-lapisan (layer) fungsi dan service.

Contoh:

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

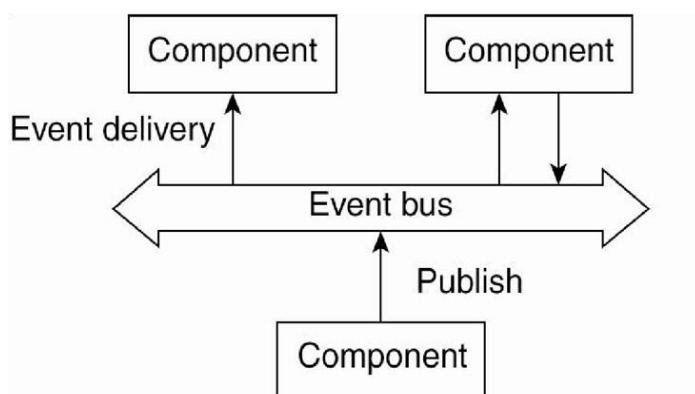
Object-base Architectures

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.



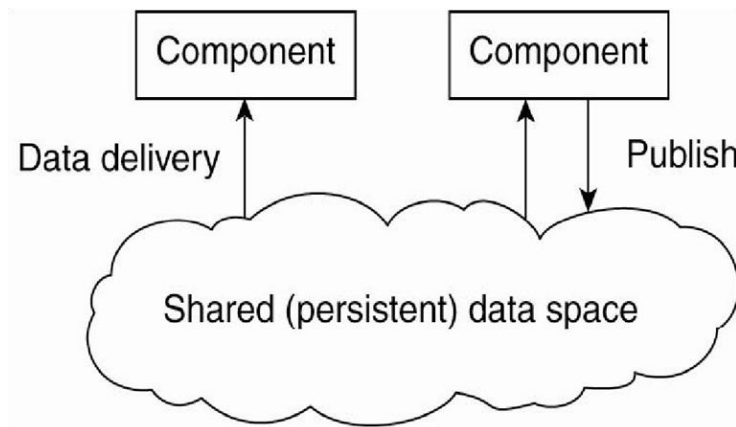
Event-based Architectures

Proses EBA pada dasarnya berdasarkan propagasi event. Proses mengeluarkan event setelah Middleware memberikan kepastian hanya proses itu saja yang bisa di subscribe untuk event yang diterima. Keuntungan EBA adalah proses bersifat loosely coupled.



Data-center Architectures

Data center dapat dipandang sebagai 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. Sebagai contoh adalah sistem tersebar



System Architecture

1) Centralized Architectures (Client-Server)

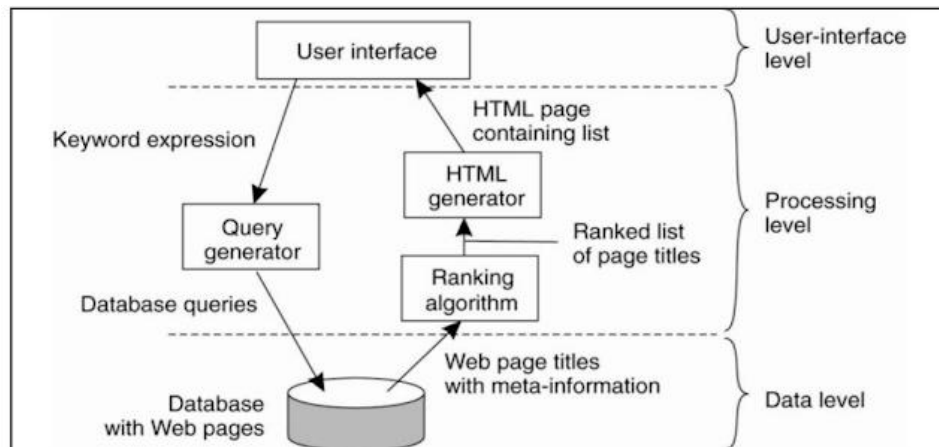
Client server unggul dalam kecepatan dan mendukung jaringan besar. Kekurangan terdapat pada sisi setup yang cukup kompleks, biaya tinggi dan membutuhkan sumberdaya manusia yang handal untuk mengelola. Pada model client server, terdapat perilaku yang biasa disebut request-reply behaviour

- Application Layering

Model client server seiring perkembangannya mengundang perdebatan mengenai perbedaan antara client dan server itu sendiri. Pada umumnya client server architecture ditujukan untuk keperluan user access ke database, maka dari itu layered architectural style dibagi menjadi:

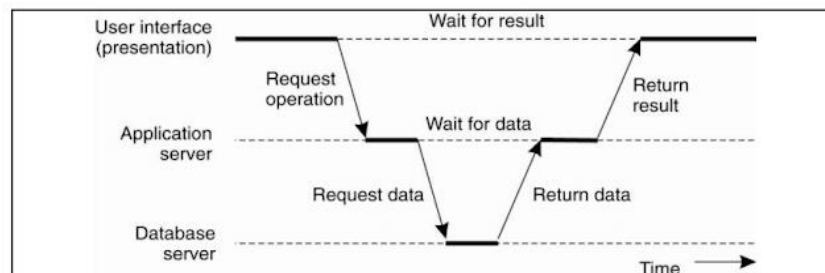
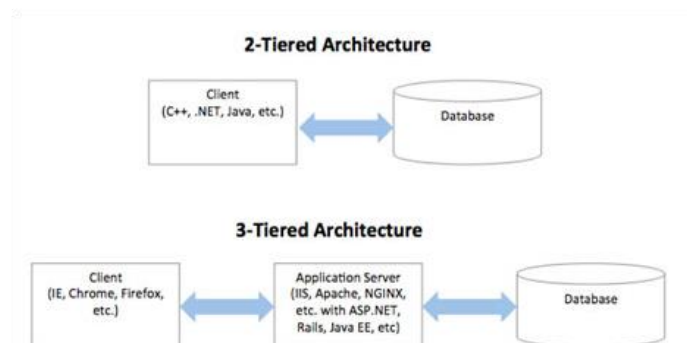
- user-interface level (display management)
- processing level (applications)
- data level (actual data that is being acted on)

Application Layering

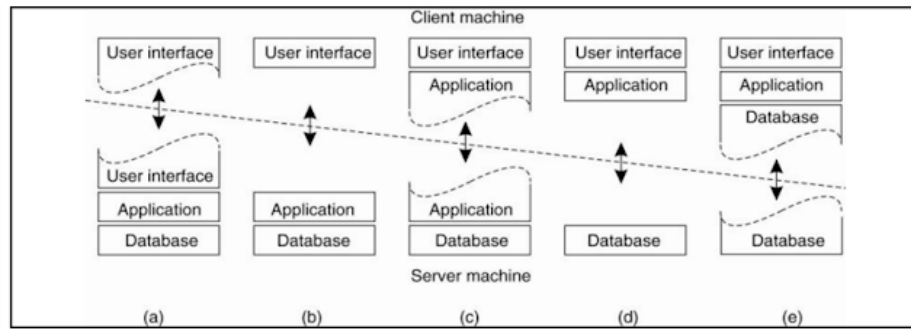


Multitier Architecture

- 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
- Pada keperluan khusus, kadang server juga perlu bertindak sebagai client
- Pada arsitektur ini, program pada processing level tidak hanya terdapat pada server yang terpisah, bahkan dapat terdistribusi pada client dan server mesin

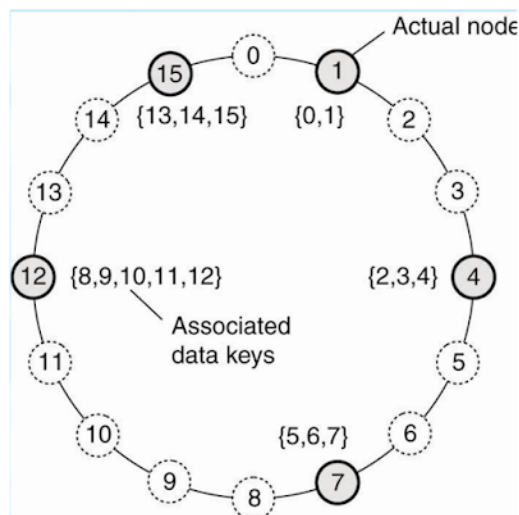


- Trend perkembangan model arsitektur

Decentralized Architecture

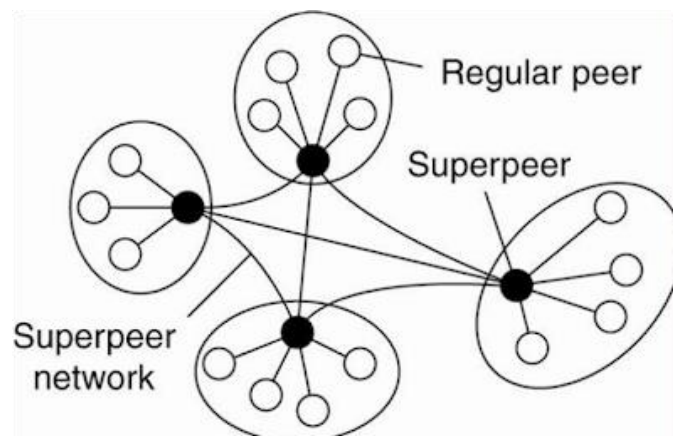
- Disebut sebagai Horizontal distribution
- Pada arsitektur ini, secara fisik terpisah namun secara logis memiliki fungsi level yang sama (equivalent), dimana setiap mesin memproses bagiannya sendiri kemudian melakukan balancing terhadap hasil proses.
- Nama lain Peer-to-peer architecture

Peer-to-peer architecture



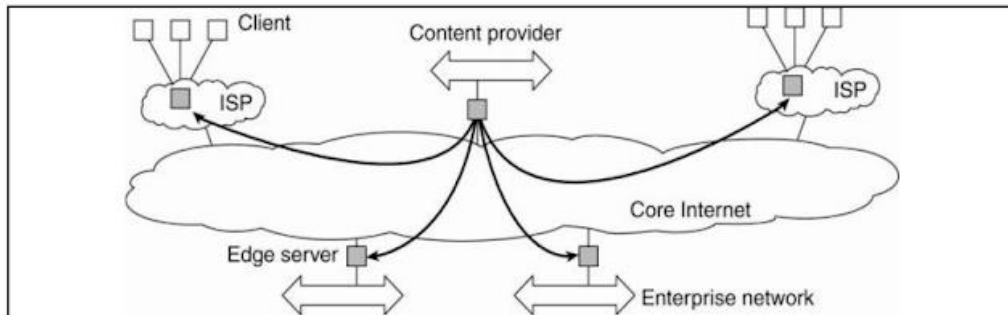
- Perkembangan P2P arsitektur tidak lepas dari pertanyaan
 1. Bagaimana organisasi proses dalam jaringan
 2. Sebuah proses tidak dapat berhubungan secara langsung dengan proses lain di jaringan.
 3. Diperlukan sebuah pesan khusus untuk komunikasi proses

- Structured peer-to-peer architecture : Dalam struktur ini lapisan jaringan di bangun menggunakan deterministic procedure, seperti menggunakan distributes hash table (DHT).
- Unstructured peer-to-peer architecture : Dalam struktur ini menugaskan sebagian besar pada algoritma secara acak untuk membangun lapisan jaringan. Pada intinya setiap node mendata jaringan node neighbor, tetapi data node tersebut di tempuh dengan proses acak sederhana.
- Topology Management of Overlay Networks : Walaupun pada structured dan unstrustured peer-to-peer System cukup jelas, namun dalam beberapa kasus masih belum lengkap. Satu kunci dari observasi adalah kehati-hatian dari proses pertukaran dan pemilihan entries dari pandangan parsial dimana topologi tertentu dapat dibangun dan dijaga konektivitasnya.
- Superpeers



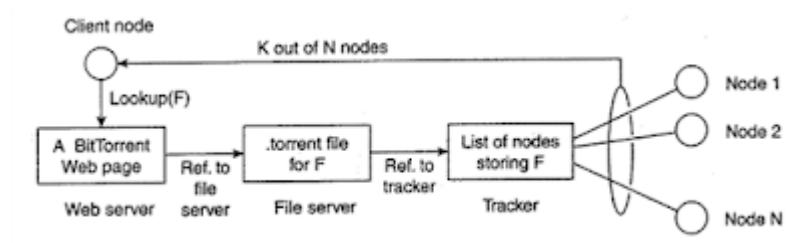
Hybrid Architecture

- 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

- Bentuk lainnya adalah CBS ini dibangun dari beberapa jaringan sistem tersebar yang ada.
- Konsep sama dengan BitTorrent file-sharing system
- Component dapat redirect client untuk akses server lain, analisa pola akses client, manage replication data

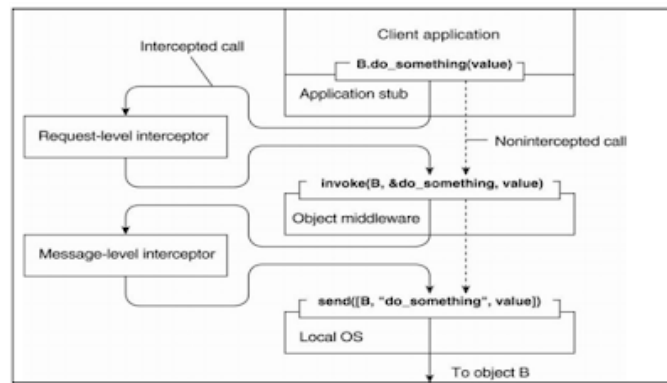


Architecture Versus Middleware

- Tidak membahas perbandingan Arsitektur dan Middleware
- Middleware mengikuti bentuk arsitektur yang ada.
- Middleware dan aplikasi menangani kebutuhan berbeda namun nantinya tetap dibutuhkan solusi dimana middleware mudah untuk di konfigurasi, disesuaikan dan di kostumisasi sesuai kebutuhan aplikasi.

Interceptors

- Interceptors merupakan perangkat lunak yang memecah aliran pengendalian dan mengijinkan kode lain untuk di eksekusi/proses.
- Interceptors sangat baik untuk menyediakan proses transparency dari Replication dan Performance



General Approaches to Adaptive Software

- Kebutuhan akan penyesuaian terhadap lingkungan aplikasi di sistem tersebar adalah perubahan secara terus menerus.
- Perubahan ini sebagai hasil dari mobility, quality-of-service networks, kerusakan hardware, dan battery drainage dll.
- Konsep ini disebut sebagai adaptive software
- McKinley et al. (2004) membagi 3 teknik dasar menuju adaptive system