

BAB III

LANDASAN TEORI

2.3. Landasan Teori

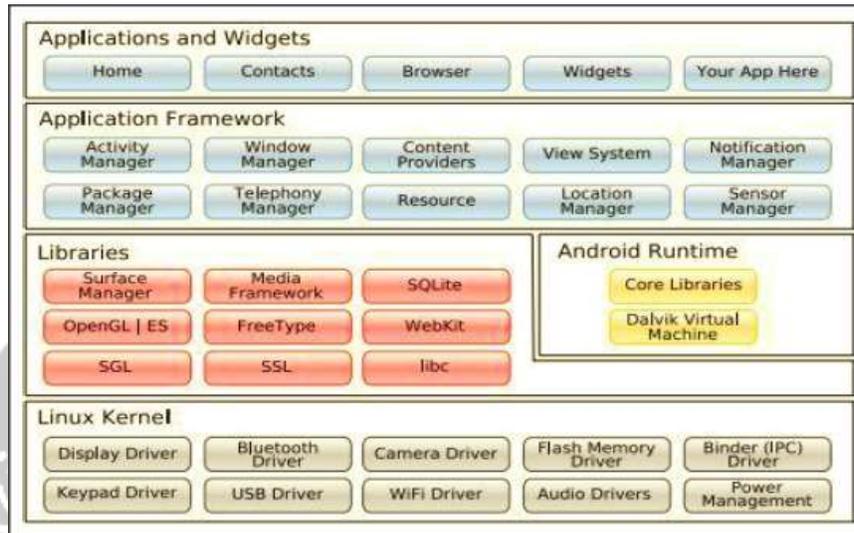
2.3.1. Android

Android adalah sistem operasi berbasis *Linux* bagi telepon seluler seperti telepon pintar dan komputer tablet. *Android* juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Safaat H, 2012).



Gambar 3.1 Logo Android

Sistem operasi Android dibangun berdasarkan kernel Linux dan memiliki arsitektur sesuai dengan Gambar 3.1.



Gambar 3.2 Arsitektur android (Safaat H, 2012)

Secara garis besar, arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

1. ***Applications dan Widgets:*** *Applications dan Widgets* ini adalah layer dimana berhubungan dengan aplikasi saja, di mana biasanya *download* aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut.
2. ***Applications Frameworks:*** *Applications frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *contact providers* yang berupa sms dan panggilan telepon.
3. ***Libraries:*** *Libraries* ini adalah *layer* di mana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan

aplikasinya. Berjalan di atas *kernel*, *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL.

4. **Android Run Time Layer:** *Android Run Time Layer* yang membuat aplikasi *Android* dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux.

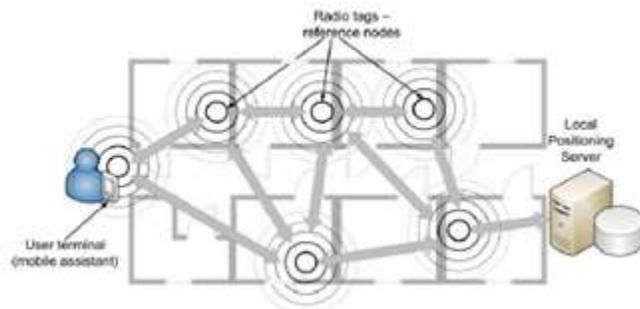
Linux Kernel: *Linux Kernel* adalah *layer* di mana inti dari operating system dari *Android* itu berada. Berisi *file-file* sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan *android* adalah linux kernel release 2.6.

2.3.2. Layanan Berbasis Lokasi

Layanan berbasis lokasi atau Location Based Service (LBS) merupakan layanan yang memberikan informasi kepada pengguna dengan didasari lokasi geografis pengguna (Lee & Yim, 2012). Saat ini banyak metode pencarian lokasi yang umum digunakan, dimana masing-masing berbeda karena memiliki kelebihan dan kelemahan masing-masing. Sebagai contoh, suatu metode bisa saja memerlukan biaya lebih besar dari yang lain namun memiliki tingkat akurasi yang lebih tinggi

3.3.3. Layanan Berbasis Lokasi Dalam Ruangan

Untuk kasus di dalam ruangan, digunakan layanan khusus yaitu layanan *Indoor Location Based Service* (ILBS).



Gambar 3.3 Skema Indoor Positioning System

Beberapa teknologi ILBS yang pernah dikembangkan (Sabour, 2007):

a. ActiveBadge (infrared)

Sistem ActiveBadge dikembangkan oleh Olivetti Research Laboratory. Sistem ini menggunakan pemancar inframerah yang melekat pada lencana pengguna. Sensor inframerah dipasang di seluruh gedung. Sebuah server pusat mengumpulkan tag unik yang dikirimkan oleh lencana pengguna dan sensor inframerah yang berada dalam ruangan.

b. Cricket (*ultrasonic*)

Penentuan lokasi Cricket menggunakan teknologi ultrasonik. Perangkat yang akan ditentukan lokasinya, melakukan triangulasi dan perhitungan time of light dalam rangka untuk mencari lokasinya. Sistem Cricket dapat mengidentifikasi dan mengabaikan sinyal ultrasonik yang lain, sehingga meningkatkan akurasi.

c. EasyLiving

Sistem Easyliving adalah sistem yang didasarkan pada analisis citra. Menggunakan Digiclops 3D kamera untuk merekam fitur dari lokasi dan membandingkannya

dengan database. Easyliving membutuhkan kamera dengan performa tinggi serta sistem yang memiliki proses yang memadai untuk menganalisa foto yang diambil dan mencocokkannya dengan database.

d. SpotOn (*radio frequency*)

SpotON adalah penentuan lokasi dengan ad hoc system. Menggunakan tag radio serta base station radio penerima sebagai dasar untuk sistem penentuan lokasi. Setiap tag radio mentransmisikan sejumlah nomor identifikasi unik. Setiap base station kemudian, mengirim nomor unik bersama dengan kekuatan sinyal yang sedang diterima dari tag ke server pusat. Server pusat kemudian menggunakan teknik tringulasi untuk menghitung lokasi dari tag.

3.3.4. Metode *location fingerprint*

Definisi *fingerprint* di sini merujuk pada lokasi pengukuran sinyal pada metode ini, untuk kasus pada metode ini sinyal yang diukur adalah *Received Signal Strength* (RSS). Dengan sistem yang mirip seperti sidik jari manusia, sidik jari lokasi yang didapatkan dengan pengukuran sinyal radio dapat digunakan untuk mengidentifikasi lokasi tertentu. Tujuan utama dari metode *fingerprint* adalah memetakan lokasi tempat pengukuran menggunakan sinyal radio di tempat yang penting (Li, 2006). Metode *fingerprint* adalah metode yang dapat digunakan untuk menentukan lokasi dengan memanfaatkan infrastruktur jaringan 802.11 WLAN (Taheri, et al., 2004).

3.3.5. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan metode dalam melakukan klasifikasi terhadap sebuah objek berdasarkan jarak yang paling dekat terhadap objek tersebut (Nasution & Mazharudin, 2011). KNN merupakan algoritma yang termasuk dalam algoritma supervised learning dimana data yang baru akan diklasifikasikan menurut data yang sudah ada pada kategori KNN. KNN juga memiliki pendekatan dalam mencari suatu kasus dengan memperhitungkan kedekatan antara suatu problem baru dengan problem sebelumnya berdasarkan dengan pencocokan bobot nilai dari beberapa fitur sebelumnya untuk dapat digunakan sebagai klasifikasi atau prediksi tergantung pada jenis data label. Tujuan dari algoritma KNN untuk mengklasifikasi objek baru berdasarkan atribut dan training samples. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada algoritma KNN.