

**PENGEMBANGAN APLIKASI MOBILE UNTUK MEMANTAU
STATUS DENYUT NADI PASIEN**

***MOBILE APPLICATION DEVELOPMENT FOR MONITORING
PATIENT STATUS PULSE***

MUHAMMAD YASSIR



**PROGRAM PASCASARJANA
UNIVERSITAS HASANUDDIN
MAKASSAR**

2013

**PENGEMBANGAN APLIKASI MOBILE UNTUK MEMANTAU
STATUS DENYUT NADI PASIEN**

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

diajukan oleh

MUHAMMAD YASSIR

Kepada

**PROGRAM PASCASARJANA
UNIVERSITAS HASANUDDIN
MAKASSAR**

2013

PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan di bawah ini :

Nama : Muhammad Yassir

Nomor Mahasiswa : P2700211024

Program Studi : Teknik Elektro

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar,

Yang menyatakan

Muhammad Yassir

KATA PENGANTAR

Dengan memanjatkan Puji Syukur kehadirat Allah SWT atas limpahan Rahmat, Karunia, HidayahNya, sehingga penulis masih diberikan nafas dan kesempatan dalam menjalani setiap proses penyelesaian tesis ini, tidak lupa Shalawat serta Salam tidak putus-putusnya penulis kirimkan kepada Junjungan Nabi Besar Rasulullah Muhammad SAW dan Keluarga serta Sahabat Beliau, Alhamdulillah penulis dapat menyelesaikan tesis ini dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Teknik pada Universitas Hasanuddin Makassar dengan judul tesis “ Pengembangan aplikasi mobile untuk memonitor status denyut nadi pasien”.

Dalam kesempatan ini dengan sepuh hati penulis mengucapkan terima kasih yang tak terhingga dan penghargaan yang setinggi-tingginya kepada :

1. Bapak / Ibu Pimpinan serta Staf Program Pascasarjana Universitas Hasanuddin Makassar
2. Bapak Muhammad Niswar, ST.,MIT.,Ph.D sebagai Ketua Komisi Pembimbing serta Bapak Amil Ahmad Ilham, ST.,MIT.,Ph.D sebagai Anggota Komisi Pembimbing yang telah meluangkan waktu, tenaga, dan pikirannya dalam membimbing penulis menyelesaikan tesis ini.
3. Bapak Dr.Ir. H. Andani Achmad, MT, Bapak Dr.-Ing. Faizal Arya Samman, MT, Dr.Eng.Intan Sari Areni, M.Eng dan Merna Baharuddin, ST.,M.Tel.,Ph.D sebagai Tim Penguji yang berkenan memberikan koreksi, penilaian dan saran terhadap tesis ini.
4. Para dosen pada Program Pasca Sarjana Teknik Informatika Jurusan Teknik Elektro Universitas Hasanuddin Makassar yang telah memberikan

banyak kontribusi dalam dan pengalaman yang berharga serta wawasan yang luas selama mengikuti proses perkuliahan.

5. Para teman teman seperjuangan sebanyak 46 orang angkatan 2011 yang kami sebut Pasca Melek'11 yang telah berbagai pengetahuan serta pengalaman suka maupun duka selama kurang lebih 2 tahun perkuliahan.
6. Ucapan terima kasih yang setinggi-tingginya secara khusus dipersembahkan kepada orang tua kami H. Nawawi HR, Hj.ST.Silmiah B, H. Andi Muhammad Siang, S.Pd dan Hj. Hafsah yang telah memberikan doa dan dukungannya selalu.
7. Adikku tercinta Roslinda, tiada kata selain terima kasih yang setinggi-tingginya.
8. Sahabat saya Suwarjono, S.Kom.,MT. dan Fitri Adi Iskandarianto, ST.,MT. yang telah banyak meluangkan waktu untuk membantu dalam proses pengerjaan tesis ini. Terima kasih banyak, semoga Allah SWT membalas kebaikannya.

Pada akhirnya penulis menyadari bahwa tesis ini masih jauh dari kesempurnaan, sehingga besar harapan penulis untuk senantiasa diberi masukan serta saran agar kedepannya menjadi lebih baik dan bermanfaat bagi pembaca.

Makassar,

12 Agustus 2013

Penulis

ABSTRAK

MUHAMMAD YASSIR. *Pengembangan Aplikasi Mobile Untuk Memantau Denyut Nadi pasien* (dibimbing oleh Muhammad Niswar dan Amil Ahmad Ilham).

Penelitian ini bertujuan untuk (1) Merancang Sistem Aplikasi *Mobile* untuk memantau status denyut nadi pasien secara *real time* menggunakan *platform android* berbasis *client server* dan (2) Mengimplementasikan aplikasi pantau denyut nadi pada pusat kesehatan masyarakat.

Penelitian ini dilaksanakan pada Rumah Sakit Umum Daerah Kabupaten Pangkep dengan menggunakan metode survey lapangan dan mengambil data kuisisioner tenaga medis sebagai responden. Penelitian dilakukan setelah sebelumnya dibuat sebuah system aplikasi *mobile* berbasis android. Aplikasi ini diuji dengan metode *functional requirement* dan angket.

Hasil pengujian menunjukkan bahwa setiap fungsi dari aplikasi ini dapat berjalan dengan baik. Sedangkan hasil penelitian menunjukkan bahwa dari 20 responden sebagian besar (95%) tenaga medis menyatakan bahwa aplikasi ini relevan untuk digunakan sebagai aplikasi memantau denyut nadi pasien. Rata-rata 85% menyatakan bahwa aplikasi ini sangat mudah digunakan serta 90% menganggap aplikasi ini sangat membantu tenaga medis.

Kata Kunci : android, *client-server*, pulse sensor, pulse monitor, *mobile application*

ABSTRACT

MUHAMMAD Yassir . Mobile Application Development For Pulse Monitor patients (led by **Amil Ahmad Ilham dan Muhammad Niswar**) .

This study aims to (1) Designing a Mobile Application System to monitor the status of the patient's pulse rate in real time using the android platform client server -based and (2) Implement a pulse monitor application in public health care .

The research was conducted at the General Hospital Pangkep using field surveys and retrieve data as a medical questionnaire respondents . The study was conducted after previously created an android based mobile application system . This application was tested by the method of functional requirements and questionnaires .

The results show that every function of this application can run well . While the results showed that the majority of the 20 respondents (95 %) medical staff stated that this application is used as an application relevant to a patient's pulse rate monitor . On average 85 % stated that this application is very easy to use and 90 % considered this application very helpful medical personnel .

Keywords : android , client - server , pulse sensor , pulse monitors , mobile application

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN	xi
DAFTAR SINGKATAN	xii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Batasan masalah	2
D. Tujuan Penelitian	3

E. Manfaat Penelitian	3
F. Sistematika Laporan	4
BAB II TINJAUAN PUSTAKA	5
A. Dasar Teori dan Konsep	5
1. Aplikasi	5
2. Basis Data	6
B. Tinjauan Hasil Penelitian Terkait	41
BAB III METODE PENELITIAN	44
A. Analisa Kondisi Awal	44
B. Usulan Rancangan Arsitektur Sistem	45
C. Alat dan Bahan	45
D. Tahapan Penelitian	46
1. Tahap Analisis Model dan Infrastruktur	47
2. Tahap Pengumpulan Data dan Studi Literatur	48
3. Tahap Desain dan Perencanaan system	49
a. Arsitektur Aplikasi	49

b. <i>Use Case Diagram</i>	50
c. <i>Activity Diagram</i>	50
d. <i>Class Diagram</i>	53
4. Tahap Implementasi dan Evaluasi Sistem	54
a. Pengujian Fungsional (Functional Requirement)	54
b. Pengujian Menggunakan Angket	54
E. Lokasi dan Waktu Penelitian	56
BAB IV HASIL DAN PEMBAHASAN	58
A. Implementasi Sistem	58
1. Gambaran Umum <i>Interconnection Thread</i> pada Software	59
2. Fungsi-fungsi Manajemen pada Server Pemantau Denyut Nadi	61
a. Edit Data Dokter	61
b. Edit Data pasien	61
c. Stream Data Pasien	61

3. Web Program Pemantau Denyut Nadi	62
a. Tampilan Antar Muka, Fungsi dan Cara Kerja	
Aplikasi Penarik data dari Sensor	62
b. Tampilan Antar Muka Web Program	63
c. Fungsi-fungsi pada Web Program “Pemantau Denyut Nadi”	63
d. Cara Kerja Web Program “Pemantau Denyut Nadi”	66
B. Aplikasi Pemantau Denyut Nadi	67
1. Tampilan Antar Muka Aplikasi	67
2. Fungsi-fungsi pada Aplikasi Pemantau Denyut Nadi	68
3. Cara Kerja Aplikasi “Pemantau Denyut Nadi”	69
C. Pengujian Sistem	70
1. Proses Menampilkan Denyut Nadi Lewat XBee	70
a. Tampilan Awal Aplikasi “Pemantau Denyut Nadi”	70
b. Tampilan Login	71
c. Tampilan Menu Dokter	71

d. Tampilan Streaming Data via XBee	72
e. Tampilan Hasil Diagnosa pasien	72
2. Proses Menampilkan Denyut Nadi Lewat Bluetooth	73
a. Tampilan Open Bluetooth	73
b. Tampilan Pengaktifan Bluetooth	73
c. Tampilan Mendeteksi Bluetooth yang Aktif	74
d. Tampilan Streaming Data via Bluetooth	74
e. Tampilan Hasil Diagnosa Dokter	75
3. Menampilkan Proses Edit Data Dokter dan Pasien	75
a. Tampilan Entry Data Dokter	75
b. Tampilan Entry Data Pasien	76
4. Menampilkan Hasil Edit Akun Dokter	76
a. Tampilan Edit Akun Dokter jika Username belum Terpakai	76
b. Tampilan Hasil Edit Akun Dokter Sudah Tersimpan	77
c. Tampilan Edit Akun Dokter jika Username sudah	

ada yang pakai	77
D. Pengujian Menggunakan Angket	78
BAB V PENUTUP	87
A. Kesimpulan	87
B. Saran	87
DAFTAR PUSTAKA	88

DAFTAR TABEL

nomor	halaman
1. Spesifikasi Kebutuhan Sistem (Overall Description)	21
2. Daftar Pertanyaan Kuesioner	52
3. Review Jumlah Responden	53
4. Hasil Kuesioner	53

DAFTAR GAMBAR

nomor	halaman
5. Logo Android	5
6. Logo Java	7
7. Logo PHP	8
8. Logo Visual basic	9
9. Tampilan panel controlxampp	11
10. Alur Sistem berjalan	18
11. Rancangan Arsitektur Sistem Usulan	19
12. Use Case Diagram untuk Aplikasi "Pantau Denyut Nadi"	24
13. Activity Login	25
14. Activity Dokter	25
15. Activity Pasien	26
16. Activity Denyut Nadi	26
17. Activity diagram diagnose	27
18. Class Diagram	27
19. Alur Penelitian	30
20. Gambaran Umum Aplikasi (Scope)	31
21. <i>System scenario</i> pada pusat kesehatan	32

22. Komunikasi Android dengan Arduino via Bluetooth	33
23. Skematik Interconnection Thread Data Software	34
24. Tampilan Respon XBee	36
25. Potongan Source Code Proses Cek Xbee	37
26. Streaming Data Salah Satu Pasien	38
27. Edit Data Dokter	39
28. Edit Data Pasien	39
29. Tampilan Menu Login Main Program Pada Web	40
30. Tampilan Awal Aplikasi Pantau "Denyut Nadi"	41
31. Grafik Hasil Kuesioner	55
32. Grafik hasil kuesioner berdasarkan jawaban responden pada pertanyaan 1	56
33. Grafik Hasil Kuesioner berdasarkan jawaban responden pada pertanyaan 2	56
34. Grafik Hasil Kuesioner berdasarkan jawaban responden pada pertanyaan 3	57
35. Grafik Hasil Kuesioner berdasarkan jawaban responden pada pertanyaan 4	58
36. Grafik Hasil Kuesioner berdasarkan jawaban responden pada pertanyaan 5	59

DAFTAR SINGKATAN

Singkatan	Keterangan
JDK	Java Development kit
JSDK	Java Software Development Kit
JRE	Java-Runtime Environment
J2SE	Java 2 Standard Edition
J2EE	Java 2 Enterprise Edition
EJBs	Enterprise JavaBeans
JSP	JavaServer Pages
Common Object Request Broker Architecture	
XML	Extensible Markup Language
J2ME	Java 2 Micro Edition
JVM	Java Virtual Machine

CLDC Configuration	Connected Limited Device
CDC	Connected Device Configuration
PDAP	Personal DigitalAssistant Profile
PDAP	Personal DigitalAssistant Profile
GMS	Google Mail Services
OHD	Open Handset Distribution
DVM	The Dalvik Virtual Machine
UI	User Interface
GPS	Global Positioning System
NFC	Near Field Communications
UML	Unified Modelling Language
OMT	Object Modeling Technique
OOSE Engineering	Object Oriented Software
IDE Environment	Integrated Development

RCP	Rich Client Platform
SWT	Standard Widget Toolkit
JDT	Java Development Tools
PDE	Plug-in Development
Environment	
API	Application Programming
Interface	
PHP	Hypertext Preprocessor
ASP	Active Server Page
PWS	Personal Web Server
IIS	Internet Information Server
JSN	Jaringan Sensor Nirkabel
SN	Sensor Node

BAB I

PENDAHULUAN

A. Latar Belakang

Jantung merupakan organ vital yang dimiliki manusia, akan tetapi banyak masyarakat yang kurang menyadari akan pentingnya kesehatan organ jantung sehingga jumlah penderita penyakit jantung semakin bertambah (Bernadeta, 2013). Berdasarkan data statistik Departemen Kesehatan Republik Indonesia pada akhir tahun 2004 mencatat 20 juta masyarakat Indonesia menderita penyakit jantung. Hasil analisa survei kesehatan rumah tangga Departemen Kesehatan RI melaporkan, penyakit *kardiovaskuler* kini menduduki jenjang tertinggi penyebab kematian. Kondisi tersebut tidak jauh berbeda dengan di negara-negara maju (Khusyu, 2013). Organisasi Kesehatan Dunia (WHO) menyebutkan rasio penderita gagal jantung di dunia satu sampai lima orang setiap 1.000 penduduk. Dengan adanya permasalahan di atas, penelitian ini berkeinginan untuk membuat suatu solusi dalam *me-monitoring* sekaligus mengantisipasi secara dini atas resiko kelainan jantung pada manusia. Solusi yang telah dibuat yaitu dengan merancang dan membuat suatu alat yang dapat memonitor keadaan detak jantung manusia. Alat ini dapat mengukur langsung denyut nadi manusia. Adapun pengembangan dari

alat ini adalah mengirim data denyut nadi tersebut ke komputer server, kemudian dari komputer server data-data dari hasil pengukuran dapat dilihat pada Smartphone. Hal ini diharapkan data monitoring tersebut bisa dilihat oleh Dokter yang berada jauh dari tempat pasien yang diukur denyut nadinya sehingga dapat memberikan diagnosa dan tindakan yang tepat terhadap pasien yang bersangkutan.

B. Rumusan Masalah

Rumusan masalah penelitian ini adalah:

1. Bagaimana cara agar pihak medis dapat terus memantau status denyut nadi pasien walaupun dokter tidak berada di sisi pasien?
2. Bagaimana hasil yang diperoleh jika aplikasi ini diimplementasikan pada sebuah pusat kesehatan masyarakat?

C. Batasan Masalah

Batasan Masalah dalam penelitian ini adalah :

- Membahas pembuatan aplikasi *android* dan *client-server*.
- Aplikasi ini hanya berjalan pada *mobile* yang memiliki profil dan konfigurasi *platform android* dan didukung jaringan *wireless*.

- Aplikasi yang digunakan untuk Data dan grafik denyut nadi di komputer dengan antarmuka berbasis web dengan script Java dan PHP dan disisi Dokter menggunakan platform android.
- Sensor yang digunakan untuk Akuisisi data menggunakan Sensor Model pulse sensor.
- Jaringan Wireless yang digunakan menggunakan Modul Xbee dan bluetooth.
- Pengujian aplikasi menggunakan *black box system*.

D. Tujuan Penelitian

Berdasarkan rumusan masalah di atas dapat disimpulkan bahwa tujuan Penelitian ini adalah :

1. Merancang Sistem Aplikasi *Mobile* untuk memantau status denyut nadi pasien secara *real time* menggunakan *platform android* berbasis *client server*.
2. Mengimplementasikan aplikasi pantau denyut nadi pada pusat kesehatan masyarakat.

E. Manfaat Penelitian

Manfaat penelitian ini adalah agar dapat memudahkan tim medis untuk mengetahui kondisi denyut nadi pasien di manapun dan kapanpun

juga, sehingga dapat memberikan diagnosa dan tindakan yang tepat terhadap pasien yang bersangkutan. Aplikasi ini juga mudah digunakan sehingga dapat dimanfaatkan untuk mengukur denyut nadi diri sendiri secara mandiri agar dapat mengantisipasi secara dini atas resiko kelainan jantung

F. Sistematika Laporan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan :

Bab 1 : Pendahuluan.

Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika laporan.

Bab 2 : Tinjauan Pustaka

Bab ini berisi literatur dan referensi yang digunakan dalam menyelesaikan penelitian ini. Bagian ini juga menjelaskan penelitian yang sebelumnya. **Bab 3: Metodologi Penelitian.**

Bab ini menjelaskan tentang perencanaan metodologi penelitian, blok diagram sistem, blok diagram perangkat lunak dan perangkat lunak yang digunakan.

Bab 4 : Analisa Data & Pembahasan.

Bab ini berisi hasil implementasi system, penjelasan tiap blok menu aplikasi serta pengujian software.

Bab 5 : Kesimpulan & Saran.

Bab ini menjelaskan tentang kesimpulan dari sistem telah dibuat berdasarkan hasil analisa data. Bab ini juga menyertakan saran untuk pengembangan ke depan.

BAB II

TINJAUAN PUSTAKA

A. Dasar Teori dan Konsep

Penyusunan penelitian ini berisi tentang pembuatan aplikasi yang dapat memantau denyut nadi pasien yang secara khusus akan diimplementasikan pada pusat kesehatan masyarakat. Aplikasi ini berjalan pada *smartphone* berbasis android dengan menarik data dari sensor denyut nadi dengan menggunakan *mikrokontroller* arduino yang kemudian diteruskan ke server. Pada bagian ini membahas teori yang mendukung isi dari penelitian ini, teknologi, android beserta *tools* yang mendukung dalam mengerjakan aplikasi ini.

1. Aplikasi

Perangkat lunak aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu *language software*, yang dapat berbentuk *assembler*, *compiler* ataupun

interpreter. Jadi *language software* merupakan bahasanya dan program yang ditulis merupakan program aplikasinya.

Language software berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Bila hendak mengembangkan suatu program aplikasi untuk memecahkan permasalahan yang besar dan rumit, maka supaya program aplikasi tersebut dapat berhasil dengan baik, maka dibutuhkan prosedur dan perencanaan yang baik dalam mengembangkannya. Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi internasional. Program-program paket tersebut dapat diandalkan, dapat memenuhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, user friendly (mudah digunakan), mempunyai dokumentasi manual yang memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya. Akan tetapi, bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri.

2. Basis Data

Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam

pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disk (disket atau hard disk). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani oleh manusia, sementara basis data dikelola/ditangani melalui perantara alat/mesin pintar elektronis (yang dikenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metoda/cara yang dapat digunakan dalam upaya penyimpanan.

2.1. Operasi Dasar Basis Data

Di dalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Di dalam sebuah disk, dapat pula ditempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file/tabel. Pada file/tabel inilah sesungguhnya data disimpan/ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya, ada basis data kepegawaian, basis data akademik, basis data inventori (Pergudangan), dan sebagainya. Sementara dalam basis data akademik, misalnya, dapat menempatkan file mahasiswa, file mata kuliah, file dosen,

file jadwal, file kehadiran, file nilai, dan seterusnya. Karena itu, operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi :

- a. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru
- b. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada).
- c. Pembuatan file/tabel dari suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
- d. Penghapusan file/tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
- e. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*), yang identik dengan penambahan ke lemari arsip ke sebuah map arsip.
- f. Pengambilan data dari sebuah file/tabel (*retrieve/search*) yang identik dengan pencarian lembaran arsip dari sebuah map arsip
- g. Pengubahan data dari sebuah file/tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.

h. Penghapusan data dari sebuah file/tabel (delete), yang identik dengan

penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

Operasi yang berkenaan dengan pembuatan objek (basis data dan tabel)

merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dan karena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (management) dan pengolahan (processing) data dalam basis data.

2.2 Objektif Basis Data

Telah disebutkan di awal bahwa tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh menemukan kembali data (yang dicari) dengan mudah dan cepat. Disamping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan-tujuan lain. Secara lebih lengkap, pemanfaatan basis

data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini :

a. Kecepatan dan kemudahan

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika menyimpan data secara manual (non-elektronis) atau secara elektronik (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk spread sheet atau dokumen teks biasa).

b. Efisiensi ruang penyimpanan (Space)

Karena keterkaitan yang erat antara kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruangan penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antara kelompok data yang saling berhubungan.

c. Keakuratan (Accuracy)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (constrain) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

d. Ketersediaan (Availability)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/refereensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi kita gunakan, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi off-line) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan *off-line* (seperti removable disk, atau tape). Di sisi lain, karena kepentingan pemakaian data, sebuah geografis. Data nasabah sebuah bank, misalnya, dipisah-pisahkan dan disimpan di lokasi yang sesuai dengn keberadaan nasabah. Dengan pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi/cabang, dapat juga di akses (menjadi tersedia/available) bagi lokasi/cabang lain.

e. Kelengkapan (Completeness)

Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relative (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama. Atau, yang sekarang dianggap sudah lengkap belum tentu di masa yang akan data juga demikian. Dalam sebuah basis data, disamping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek seperti struktur file/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambahkan record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

f. Keamanan (Security)

Memang ada sejumlah sistem (aplikasi) pengelola basis data yang tidak

menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu kita dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek

di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

g. Kebersamaan Pemakai (Sharebility)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja, atau di satu lokasi saja atau boleh satu sistem/aplikasi saja, data pegawai dalam basis data kepegawaian, misalnya, dapat digunakan oleh banyak pemakai, dari sejumlah departemen dalam perusahaan atau oleh banyak sistem (sistem penggajian, sistem akuntansi, sistem inventori, dan sebagainya). Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi deadlock(karena banyak pemakai yang saling menunggu untuk menggunakan data).

2.3 Java

2.3.1 Pengertian Java

Java adalah bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya seperti Patrick Naughton, Chris

Warth, Ed Frank, dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama Sun Microsystem, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “oak”, namun pada tahun 1995 diganti namanya menjadi “Java” Alasan utama pembentukan bahasa java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan diberbagai macam perangkat elektronik, seperti microwave oven dan remote control, sehingga Java harus bersifat portable atau yang sering disebut dengan platform independent (tidak tergantung pada platform). Itulah yang menyebabkan dalam dunia pemrograman Java, dikenal adanya istilah ‘write once, run everywhere’, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan dibawah platform manapun, tanpa harus melakukan perubahan kode program.

2.3.2 Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun semenjak awal mula bahasa tersebut dirilis. Kompiler Java (yang disebut dengan Javac atau Java Compiler) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu bytecode. Apa itu bytecode? Bytecode adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (Java Virtual Machine). JVM juga sering dinamakan sebagai interpreter, karena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam *bytecode* dengan cara baris demi baris.

2.3.3 Java Versi Lama (Java 1)

Pada awal perilisannya, versi Java masih disebut dengan JDK (Java Development kit). Dalam JDK, semua kebutuhan untuk pengembangan program dan eksekusi program masih tergabung jadi satu. Penamaan ini berlaku sampai Java 1.1 namun sekarang, setelah Java 1.2, Sun Microsystem menamainya dengan JSDK (Java Software Development Kit) dalam hal ini kebutuhan untuk pengembangan program dipisahkan dengan kebutuhan eksekusi. Bagian software yang digunakan untuk kebutuhan eksekusi program disebut dengan JRE (Java-Runtime Environment). Selanjutnya, Java 1.2 disederhanakan penamaanya menjadi "Java 2".

2.3.4 Java 2

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2, yaitu sebagai berikut :

- a. Java 2 Standard Edition (J2SE), yang digunakan untuk mengembangkan aplikasi-aplikasi desktop dan applet (aplikasi Java yang dapat dijalankan di dalam browser web).

b. Java 2 Enterprise Edition (J2EE), merupakan superset dari J2SE yang

memperbolehkan untuk mengembangkan aplikasi-aplikasi berskala besar (enterprise), yaitu dengan melakukan pembuatan aplikasi-aplikasi di sisi server dengan menggunakan EJBs (Enterprise JavaBeans), aplikasi web dengan menggunakan Servlet dan JSP (JavaServer Pages) dan teknologi lainnya seperti CORBA (Common Object Request Broker Architecture) dan XML (Extensible Markup Language).

c. Java 2 Micro Edition (J2ME), merupakan subset dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh

2.3.5 J2ME

J2ME merupakan sebuah kombinasi yang terbentuk antara kumpulan interface Java yang sering disebut dengan Java API (Application programming interface) dengan JVM (Java Virtual Machine) yang didesain khusus untuk alat, yaitu JVM dengan ruangan terbatas. Kombinasi tersebut kemudian digunakan untuk melakukan pembuatan

aplikasi-aplikasi yang dapat berjalan di atas alat (dalam hal ini mobile device). J2ME sendiri pada dasarnya terdiri dari tiga buah bagian, yaitu konfigurasi, profil, dan paket-paket opsional, seperti yang ditunjukkan oleh gambar 2.1.



Gambar 2.1 Bagian-bagian di dalam platform J2ME

2.3.5.1 Konfigurasi

Konfigurasi merupakan bagian yang berisi JVM dan beberapa library kelas lainnya. Perlu diperhatikan bahwa JVM yang dimaksud disini bukanlah JVM tradisional seperti yang terdapat dalam J2SE, melainkan JVM yang sudah didesain secara khusus untuk alat. Terdapat dua buah konfigurasi yang disediakan oleh Sun Microsystems, yaitu CLDC (Connected Limited Device Configuration) dan CDC (Connected Device Configuration). Target alat dari konfigurasi CLDC adalah alat-alat kecil, seperti telepon selular, PDA, dan pager. Membahas lebih jauh mengenai

konfigurasi CLDC pada bagian tersendiri dalam bab ini. Pada sisi yang lain, CDC merupakan superset dari CLDC sehingga semua yang kelas yang didefinisikan di dalam CLDC akan ada juga didalam CDC.

2.3.5.2 Profil

Profil merupakan bagian perluasan dari konfigurasi. Artinya, selain sekumpulan kelas yang terdapat pada konfigurasi, terdapat juga kelas-kelas spesifik yang didefinisikan lagi di dalam profil. Dengan kata lain, profil akan membantu secara fungsional yaitu dengan menyediakan kelas-kelas yang tidak terdapat di level konfigurasi. Adapun profil yang sangat populer penggunaannya adalah profile yang disediakan oleh Sun Microsystems, yaitu yang dinamakan MIDP (Mobile Information Device Profile). Berikut ini beberapa profil yang tersedia untuk kebutuhan-kebutuhan spesifik lainnya.

a. PDAP (Personal Digital Assistant Profile) yaitu profil untuk PDA yang

memperluas fungsi-fungsi pada konfigurasi CLDC dan digunakan khusus untuk menambahkan kemampuan-kemampuan lebih apabila dibandingkan dengan penggunaan profil MIDP.

b. Foundation Profile, yaitu profil yang digunakan untuk konfigurasi CDC. Profil ini menambahkan beberapa kelas dari J2SE ke dalam konfigurasi

CDC, dan berperan juga sebagai pondasi untuk membentuk profil baru lainnya.

- c. Personal Profile, yaitu profil yang mendefinisikan ulang Personal Java sebagai profil yang dapat digunakan sebagai profil dalam J2ME, Profil ini merupakan hasil perluasan dari Foundation Profile.
- d. RMI Profile, yaitu profil yang menambahkan dukungan RMI (Remote Method Invocation) ke dalam konfigurasi CDC.

2.3.5.3 Paket-paket Opsional

Paket-paket opsional merupakan paket-paket tambahan yang dibutuhkan oleh aplikasi sehingga pada saat proses deployment paket-paket tersebut perlu didistribusikan juga sebagai bagian dari aplikasi bersangkutan. Sebagai catatan bahwa paket-paket opsional ini bukan merupakan paket yang dibuat oleh perusahaan alat yang digunakan.

2.4 Android

Pembahasan mengenai android dan tools yang digunakan dalam pembuatan aplikasi, akan dijelaskan pada sub bab berikut.

2.4.1 Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, Htc, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama

Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan android sebagai sistem operasinya. Telpon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android AEM Holdings, Atheros Communication, diproduksi oleh Asustek Komputer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan

produk perdana mereka, android perangkat mobile yang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan vendor-vendor smartphone sudah memproduksi smartphone berbasis android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendor smartphone didunia yang memproduksi android. Hal ini karena android adalah sistem operasi yang open source sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di smartphone, saat ini android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena android itu sendiri adalah platform sangat lengkap baik itu sistem oeprasinya, aplikasi dan Tool Developmen, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas Open source didunia, sheingga android terus berkembang pesat dari segi teknologi maupun dari segi jumlah device yang ada didunia.

2.4.2 The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari android adalah Dalvik Virtual Machine (DVM). Android berjalan di Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaan dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi android menggunakan Virtual Machine sendiri yang diskostumisasi dan dirancang untuk memastikan bahwa beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile. Dalvik Virtual Machine (DVM) adalah “register bases” sementara java Virtual Machine (JVM) bersifat “stack based”, DVM didesain dan ditulis oleh Dan Bornsten dan beberapa enginers Google lainnya. Jadi bisa dikatakan “Dalvik equal(Java) == False” Dalvik Virtual Machine menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, threading, dan proses

serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C/C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua hardware yang berbasis android dijalankan dengan menggunakan

Virtual Machine untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. Dalvik Virtual Machine mengeksekusi Executable file, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. The Executable file diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

2.4.3. Arsitektur Android

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut :

a. Application and Widgets

Application and widgets adalah layer dimana berhubungan dengan aplikasi dan biasanya download aplikasi kemudian lakukan instalasi dan

jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. Application Frameworks

Android adalah "Open Development Platform" yaitu android menawarkan

kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan serive background, mengatur alarm, dan menambahkan tambahan seperti status notifications dan masih banyak lagi. Pengembang memiliki akses penuh menuju API Framework seperti yang dilakukan oleh aplikasi yang kategori inti.

Arsitektur aplikasi dirancang supaya dengan mudah dapat menggunakan

komponen yang sudah digunakan (reuse). Sehingga bisa disimpulkan Application Framework adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang, seperti content-providers yang berupa sms dan lain sebagainya.

Komponen-komponen yang termasuk didalam application Framework adalah sebagai berikut :

- 1) Views
- 2) Content Provider
- 3) Resource Manager
- 4) Notification Manager
- 5) Activity Manager

c. Libraries

Libraries adalah layer dimana fitur-fitur android berada biasanya para pembuat aplikasi kebanyakan mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

- 1) Libraries media untuk pemutar media audio dan video
- 2) Libraries untuk manajemen tampilan
- 3) Libraries Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D
- 4) Libraries SQLite untuk dukungan database

5) Libraries SSL dan WebKit terintegrasi dengan web browser dan security

6) Libraries Live Webcore mencakup modern web browser dengan engine

embedded web view

d. Android Run Time Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam Android Run Time dibagi menjadi dua bagian yaitu:

1) Core Libraries : aplikasi android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesin bukan Java Virtual Machine, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang dihandle oleh Core Libraries

2) Dalvik Virtual Machine : Virtual mesin yang berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangan yang mampu membuat linux kernel untuk threading dan manajemen tingkat rendah.

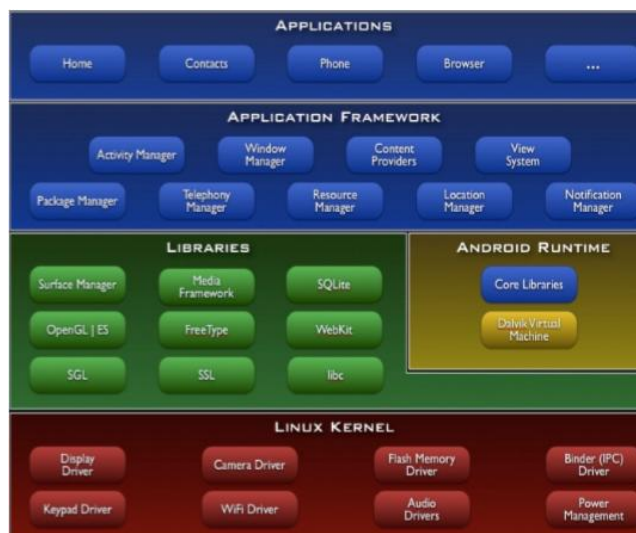
3) Linux kernel

Linux kernel adalah layer dimana inti dari operating sistem dari android itu

sendiri, berisi file-file sistem yang mengatur sistem processing, memory,

resources, drivers, dan sistem-sistem operanting android lainnya.

Linux kernel yang digunakan android adalah linux kernel release 2.6.



Gambar 2.2 Arsitektur Andoroid

2.4.4 Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java, kode java dikompilasi bersama data file resource yang dibutuhkan oleh aplikasi dimana prosesnya dipaket oleh tools yang dinamakan "apt tools" ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang sebenarnya disebut dengan aplikasi yang dapat diinstal di perangkat mobile nantinya. Ada enam jenis komponen pada aplikasi android yaitu:

a. Activities

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain dapat dilakukan dengan satu even misalnya click tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah *window activity* dinyatakan dengan method `Activity setContentView()`.

ContentView adalah objek yang berada pada root hirarki.

b. Service

Service tidak memiliki visual user interface (UI), tetapi service berjalan secara background, sebagai contoh dalam memainkan music, service mungkin memainkan music atau mengambil data dari jaringan, tetapi setiap service haruslah berada dalam kelas induknya. Misalnya media player sedang memutar lagu dari list yang ada. Aplikasi ini akan memiliki dua atau lebih activity yang memungkinkan user untuk memilih lagu atau menulis sms sambil player sedang jalan untuk menjaga music tetapi dijalankan, activity player dapat menjalankan service untuk membuat aplikasi tetap berjalan. Service dijalankan pada thread utama dari proses aplikasi.

c. Broadcast Receiver

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu berubah, baterai low, gambar telah selesai diambil oleh kamera, atau pengubah referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi broadcast misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah didownload ke perangkat dan siap untuk digunakan Broadcast Receiver tidak memiliki user interface (UI) tetapi memiliki sebuah activity untuk merespon informasi yang mereka terima atau mungkin menggunakan notification manager untuk memberitahu kepada pengguna seperti lampu latar atau viberating (getaran) perangkat dan lain sebagainya.

d. Content Provider

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite. Content Provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu activity, misalnya ketika menggunakan aplikasi yang membutuhkan peta (MAP) atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi content provider.

2.4.5. Versi Android

Telepon pertama yang memakai sistem operasi android adalah HTC Dream yang dirilis pada 22 Oktober 2008. Pada penghujung 2010 diperkirakan hampir semua vendor seluler didunia menggunakan android sebagai operating sistem. Adapun versi-versi android yang pernah dirilis adalah sebagai berikut :

a. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. android versi ini dilengkapi dengan pembaruan estis pada aplikasi, jam, alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

b. Android versi 1.5 (cupcake)

Pada pertengahan 2009, google kembali merilis telepon seluler dengan

menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengupload video Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, dan keyboard pada layer yang dapat disesuaikan oleh sistem.

c. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik disbanding sebelumnya, pengguna baterai indicator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang diintegrasikan, CD/EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine, kemampuan dial kontak, teknologi text to change speech (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

d. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Éclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3,1,2, perubahan UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan flash untuk kamera 3.2 MP, digital Zoom, dan Bluetooth 2.1.

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada bulan Mei 2010 android 2.2 Rev 1 diluncurkan. Android inilah yang sangat banyak sekarang dipasaran, salah satunya adalah dipakai di Samsung FX tab yang sudah ada dipasaran. Fitur yang tersedia di android versi ini sudah kompleks sekali diantaranya adalah :

- 1) Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia
- 2) Dalvik Virtual Machine dioptimalkan untuk perangkat mobile.
- 3) Grafik: grafik di 2D dan 3D berdasarkan libraries OpenGL
- 4) SQLite: untuk penyimpanan data.
- 5) Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- 6) GSM, Bluetooth, EDGE, 3G, dan WiFi (tergantung Hardware)

7) Kamera, Global Positioning System (GPS), kompas, dan accelerometer

(tergantung Hardware)

f. Android versi 2.3 (Gingerbread)

Android versi 2.3 diluncurkan pada desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan sebagai berikut :

- 1) SIP-based VoIP
- 2) Near Field Communications (NFC)
- 3) Gyroscope dan sensor
- 4) Multiple cameras support
- 5) Mixable audio effects
- 6) Download manager

Itulah beberapa versi android yang sudah dirilis sampai sekarang, kemungkinan besar versi tersebut akan terus berkembang seiring dengan kebutuhan yang sangat kompleks dibidang penggunaan smartpone. Untuk mendapatkan informasi versi terbaru android dapat dilihat di www.developer.android.com.

2.5 Tools-Tools yang digunakan

2.5.1 Konsep Perancangan Berorientasi Obyek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata

yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi obke. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata.

Suatu perusahaan software yaitu Rational Software, telah membentuk

konsarium dengan berbagai organisasi untuk meresmikan pemakaian Unified Modelling Language (UML) sebagai bahasa standar dalam Object Oriented Analysist Design (OOAD).

2.5.1.1 Unified Modelling Language (UML)

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi

obyek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, Object Modeling Technique (OMT) dan Object Oriented Software Engineering (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode Design Object Oriented. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih member penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Design UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya.

2.5.1.2 Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use Case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut scenario. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu.

Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna. Dalam pembicaraan tentang use case, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model use case adalah bagian dari model requirement (Jacob et all, 1992). Termasuk disini adalah problem domain object model dan penjelasan tentang user

interface. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

2.5.1.3 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan message yang diletakkan diantara obyek-obyek ini di dalam use case. Sequence diagram menambahkan dimensi waktu pada interaksi diantara obyek. Pada diagram ini participant diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. Life line participant diurutkan dari setiap participant. Kotak kecil pada lifeline menyatakan activation, yaitu menjalankan salah satu operation dari participant. State bisa ditambahkan dengan menambahkannya sepanjang life line.

Message (sederhana, synchronous atau asynchronous) adalah tanda panah yang menghubungkan suatu life line ke life line yang lain. Lokasi life line dalam dimensi vertikal mewakili urutan waktu dalam sequence diagram. Message yang pertama terjadi adalah yang paling

dekat dengan bagian atas diagram dan yang terjadi belakangan adalah yang dekat dengan bagian bawah.

Pada beberapa sistem, operasi bisa dilakukan kepada dirinya sendiri. Hal ini disebut dengan rekursif. Untuk melukiskannya digunakan anak panah dari activation kembali ke dirinya sendiri, dan sebuah kotak kecil diletakkan pada bagian atas dari activation.

2.5.1.4 Collaboration Diagram

Collaboration Diagram adalah bentuk lain sequence diagram. Bila sequence diagram diorganisir menurut waktu maka collaboration diagram diorganisir menurut ruang/space. Collaboration diagram merupakan asosiasi diantara obyek-obyek. Panah di dekat garis asosiasi menunjukkan message, sedangkan content message ditunjukkan dengan label. Angka pada message menunjukkan urutan message Dengan collaboration diagram memungkinkan untuk memodelkan pengiriman sebuah message ke banyak obyek pada class yang sama. Demikian juga

halnya untuk menunjukkan adanya obyek aktif yang mengendalikan aliran dari message.

2.5.1.5 Activity Diagram

Activity diagram seperti sebuah flow chart. Activity diagram menunjukkan

tahapan, pengambilan keputusan dan percabangan. Diagram ini sangat berguna untuk menunjukkan operation sebuah obyek dan proses bisnis. Kelebihan activity diagram dibandingkan flowchart adalah kemampuannya dalam menampilkan aktivitas parallel.

2.5.1.6 Component Diagram

Component diagram merepresentasikan dunia rill item yaitu component

software. Component software menetap di komputer bukan di benak para analis. Component bisa diakses melalui interface nya yaitu koleksi operasi-operasi. Relasi antara component dan interfacenya disebut realization. Suatu component bisa mengakses service-service yang ada di component lain dengan cara import interface. Sedangkan component yang menyediakan service menggunakan export interface.

Hal penting dari component adalah component mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. Dengan demikian pembagian sistem ke dalam component lebih didorong dengan kepentingan marketing daripada kepentingan teknis.

2.5.1.7 Deployment Diagram

Deployment diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem terdiri dari node-node dimana setiap node diwakili untuk sebuah kubus. Garis yang menghubungkan antara 2 kubus menunjukkan hubungan diantara kedua node tersebut. Tipe node bisa berupa device yang berwujud hardware dan bisa juga processor (yang mengeksekusi component) atau execution environment (software yang menjadi host atau mengandung software yang lain).

2.5.1.8 Class Diagram

Kotak adalah notasi UML untuk class. Nama, attribute, operation dan responsibility dari class ada pada kotak tersebut. Stereotype bisa dipergunakan untuk mengorganisasikan daftar attribute dan operation. Dalam beberapa kasus, kadangkala hanya perlu ditampilkan sebagian saja dari attribute dan operation. Tipe attribute dan nilai default bisa dimunculkan sebagaimana pada operation. Untuk mengurangi ambiguitas

pada pendeskripsian class, constraint bisa ditambahkan. Bahkan kalau perlu bisa ditambahkan attached notes ke dalam kotak tersebut.

2.5.2 Eclipse

2.5.2.1 Pengertian Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk

mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

a. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

b. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

c. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini

merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

2.5.2.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang

mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

- a. Core platform
- b. OSGi
- c. SWT (Standard Widget Toolkit)
- d. JFace
- e. Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development

Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools).

Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.5.3 Android SDK

Android SDK adalah tools API (Application Programming Interface) yang

diperlukan untuk mulai mengembangkan aplikasi pada platform android

menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Hadphone/Smartphone. Beberapa fitur-fitur android yang paling penting adalah :

a. Framework : aplikasi yang mendukung pengganti komponen dan reusable.

b. Dalvik Virtual Machine dioptimalkan untuk perangkat mobile

c. Integrated Browser berdasarkan engine open source WebKit.

d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi hardware)

e. SQLite untuk penyimpanan data.

f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung Hardware)

g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)

h. Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware)

i. Lingkungan Development yang lengkap dan termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse. Untuk source SDK Android ini dapat dilihat dan didownload langsung di situs resmi pengembang SDK Android di <http://www.developer.android.com>.

2.5.5 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal, kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu, ia bersifat Open source (Anda tidak perlu membayar untuk menggunakannya) pada pelbagai platform (kecuali untuk jenis enterprise, yang bersifat komersial). Perangkat lunak MySQL sendiri bisa di download dari <http://www.mysql.com> MySQL termasuk jenis RDBMS (Relational Database Management). Itulah sebabnya, istilah seperti tabel, baris, dan

kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

2.5.6 PHP

Menurut dokumen resmi PHP, PHP merupakan singkatan dari PHP Hypertext Preprocessor. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnya yang dikirim ke klien tempat pemakai menggunakan browser.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis.

Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (Active Server Page), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara command line. Artinya, skrip PHP dapat dijalankan tanpa melibatkan web server maupun browser.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi tool yang disebut "Personal home page". Paket inilah

yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan

kompleks sambil jalan.

Pada saat ini PHP cukup populer sebagai peranti pemrograman Web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada server-server yang berbasis UNIX, Windows, dan Machintosh. Pada awalnya, PHP dirancang untuk diintegrasikan dengan web server Apache. Namun, belakangan PHP juga dapat bekerja dengan web server seperti PWS (Personal Web Server), IIS (Internet Information Server), dan Xitami.

Untuk mencoba PHP, anda tidak perlu menggunakan komputer bekelas

server. Hanya dengan sebuah Komputer biasa, anda bisa mempelajari dan mempraktikkan PHP. Pemrograman berorientasi objek merupakan model pemrograman yang menjadi tren saat ini. PHP juga mendukung hal ini. Suatu objek dapat dibentuk melalui kelas. Dalam hal ini, kelas dapat dianggap sebagai cetakan objek. Jika menggunakan istilah teknis, kelas adalah definisi umum untuk sekelompok objek. Objek

sendiri sering disebut instan kelas. Dengan menggunakan cetakan, anda bisa menciptakan objek.

B. Tinjauan Hasil Penelitian Terkait

- 1. Andi Gita Novianti.** Universitas Sain dan Teknologi Jayapura. Pembuatan Aplikasi Server Cloud Menggunakan Sistem Operasi Mobile Berbasis Android, 2012.

Umumnya, seorang administrator jaringan dalam melaksanakan tugasnya untuk mengontrol dan memonitoring server serta mengawasi aktivitas user, dilakukan dengan menggunakan sebuah komputer desktop yang terhubung dengan server dan computer yang digunakan oleh client. Perangkat tersebut biasanya diletakkan di dalam ruangan khusus untuk admin. Kondisi demikian mengharuskan admin untuk berada didalam ruangan untuk bisa melakukan monitoring dan kontrol terhadap server. Hal ini tentu saja akan sedikit merepotkan, karena tidak mungkin seorang admin akan berada di ruang admin sepanjang hari.

Untuk mengatasi masalah ini, dibutuhkan sebuah tools untuk bisa mengatasi permasalahan tersebut. Untuk membantu pelaksanaan tugas dari seorang administrator jaringan, dirancang sebuah aplikasi yang bisa membantu seorang admin untuk terus

mengontrol dan memonitoring server cloud secara jarak jauh (remote). Aplikasi ini dapat membantu admin melakukan semua aktivitas harian yang berkaitan dengan manajemen server cloud antara lain manajemen storage, manajemen instance, dan manajemen image, sekalipun admin tidak berada di dalam ruang server melalui gadget smartphone berbasis android.

- 2. Himawan.** Institut Teknologi Surabaya. Perancangan dan Implementasi Aplikasi Pemesanan Makanan dan Minuman Berbasis Client-Server Dengan Platform Android, 2011.

Aplikasi ini memakai computer di sisi server dapur dan kasir *mobile android* di sisi *client* pemesanan (pelayan). Dimana ketika pelayanan menerima pesanan makanan dari pengunjung akan di *input* di *mobile android* dan *view* di bagian dapur dan kasir. Bahasa pemrograman yang membaca inputan dari sisi *client* di dapur menggunakan *script* PHP. *Script* PHP menerima data dan selanjutnya akan membangkitkan perintah MySQL dalam bentuk *query* dan memanggil *stored procedure* yang ada pada server basis data. Hal ini bertujuan memberikan solusi alternative untuk pemesanan makanan dan minuman dengan menggunakan *mobile android*.

- 3. Y. Zhang dan H. Xiao,** *Bluetooth-based sensor networks for remotely monitoring the physiological signals of a patient, WPAN Transactions on Information Technology in Biomedicine*, vol 13, no.6, pp. 10401048, 2009.