

APLIKASI SCHEDULE PLANNING MENGGUNAKAN PLATFORM ANDROID

Shiddieqy Y. Katili

Nur Heri Cahyana, S.T.,M.KOM, Yuli Fauziah, S.T.,M.T

Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional "Veteran"
Yogyakarta

Email : shiddieqy_katili@yahoo.com

Abstrak - Untuk mencapai suatu efisiensi dan keakuratan data dalam proses manajemen waktu dan kegiatan, diperlukanlah sebuah sistem yang mampu memfasilitasi dan dapat berinteraksi agar proses manajemen waktu menjadi lebih efektif dan dapat terlaksana dengan baik Teknologi Smartphone dan tablet PC (Personal Computer) yang bersifat mobile sangat memungkinkan dan mendukung kebutuhan tersebut dapat tercapai.

Untuk perencanaan jadwal kegiatan menggunakan Metode Program Evaluation and Review Technic (PERT) dan Critical Path Methode (CPM) dalam estimasi waktu, dan menggunakan Activity On Arrow sebagai pendekatan analisa jaringan kerja. Untuk penelitian sendiri menggunakan metode pengembangan Unified Software Developmet Process (USDP) dan metode Unified Software Developmet Process (UML) proses untuk membuat perancangan sistem.

Aplikasi Schedule Planning menggunakan platform android ini, dapat mempermudah user dalam merencanakan sebuah tasklist dari sebuah project agar lebih efektif dan efisien. Selain itu user dapat membuat sebuah agenda rutin dan memo sehingga aktifitas user lebih tertata rapi.

Kata Kunci :Android, PERT-CPM, USDP, UML, Jaringan Kerja, Lintasan Kritis

Abstract - To reach an efficiency and accuracy of the data within the management process time and activities, requires the a system that is able to facilitate the process and be able to interact in order to become more effective time management can be implemented properly and Technology Smartphones and tablet PC (Personal Computer) that are highly mobile enable and support those needs can be achieved.

To schedule planning activities using Method Program Evaluation and Review Technic (PERT) and Critical Path Method (CPM) in the estimation of time, and using Activity On Arrow as network analysis approach. To own research using methods developed Unified Software Developmet Process (USDP) and Unified Software Developmet Process method (UML) to make the system design process.

Schedule Planning application using the android platform, can facilitate a user in planning a tasklist of a project to make it more effective and efficient. In addition, the user can create a routine agenda and memo so that a more organized user activity.

Keywords:Android, PERT-CPM, USDP, UML, network, critical path

1. PENDAHULUAN

Teknologi *Smartphone* dan *tablet PC* (*Personal Computer*) yang bersifat *mobile* sangat memungkinkan dan mendukung kebutuhan user yang perlu perencanaan kegiatan yang efisien. Pada setiap *smartphone* android sebenarnya telah disertakan aplikasi penjadwalan seperti *memo* dan *callender*. Pada aplikasi tersebut sebenarnya hampir semua penjadwalan kerja dapat dilakukan lebih mudah. Tetapi kedua aplikasi tersebut tidak sinkron karena tidak terdapat dalam satu aplikasi yang sama. Selain itu aplikasi tersebut hanya membandingkan waktu belum menggunakan metode penjadwalan yang ada sehingga optimasi dan efisiensi pada penjadwalan tersebut belum bisa tercapai.

Perencanaan jadwal kegiatan merupakan suatu upaya untuk mempermudah dalam manajemen waktu agar lebih terorganisir dan efektif. Banyak hal perencanaan jadwal misalnya *tasklist*, *memo*, *meeting*, dan *event plan* tetapi proses pencatatannya masih menggunakan buku/agenda, adapun pencatatan sudah menggunakan komputer tetapi masih belum optimal penggunaannya karena belum menggunakan metode penjadwalan, sehingga nilai optimal dari kegiatan belum dapat tercapai. Untuk itu Aplikasi *Schedule Planning* ini merupakan pengembangan dari aplikasi yang sudah pernah ada pada *smartphone* android yang mencakup semua perencanaan jadwal kegiatan dengan menggunakan Metode *Program Evaluation and Review Technic* (PERT) dan *Critical Path Methode* (CPM) dalam estimasi waktu dan penggunaan biaya yang lebih ekonomis, dan menggunakan *Activity On Arrow* sebagai pendekatan analisa jaringan kerja.

Dari pemaparan diatas maka digunakanlah *Unified Software Developmet Process* (USDP). Metode *Unified Software Developmet Process* merupakan salah satu metode perancangan yang menjadi dasar dari perencanaan, perancangan, dan pembangunan aplikasi. Metode *Unified Software Developmet Process* menggunakan *Unified Modeling Language* (UML) sebagai sarana pengembangan sistem/perangkat lunak yang sempurna terhadap kebutuhan dan harapan *user* serta dari sudut pandang sistem/perangkat lunak yang memiliki cacat yang minimal dengan konsep tahapan *Object-oriented analysis* dan *Object-oriented design*.

2. TEORI

Aplikasi adalah merupakan program yang berisikan perintah-perintah untuk melakukan

pengolahan data (Hartono, 2005). Dari definisi tersebut maka dapat ditarik kesimpulan pengertian aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program untuk menyelesaikan tugas-tugas khusus sehingga data yang diolah lebih berdaya guna secara optimal.

Schedule planning merupakan dua buah kata bahasa inggris yang terdiri dari *schedule* dan *planning* dimana dalam bahasa Indonesia dapat diartikan penjadwalan dan perencanaan. Penjadwalan adalah proses pengalokasian sumber-sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu (Baker, 1974). Perencanaan adalah pemilihan atau penetapan tujuan organisasi dan penentuan strategi, kebijakan proyek, program, prosedur, metode, sistem, anggaran dan standar yang dibutuhkan untuk mencapai tujuan (Handoko, 1995).

2.1. Pengertian dan komponen Android

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang di *release* oleh Google (Mulyadi, 2010).

1. *Activities*. Merupakan bagian yang paling penting dalam aplikasi *android*, *activity* normalnya akan menyajikan UI bagi pengguna.
2. *Service*. Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.
3. *Intents*. *Intents* merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman *web*, dan lain sebagainya.
4. *Broadcast reciever*. *Broadcast receiver* merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan (*notification*). *Broadcast* sebagian berasal dari sistem, misalnya baterai sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default* pada perangkat. *Service* sama halnya, *broadcast receivers* tidak menampilkan antarmuka pengguna.
5. *Content providers* digunakan untuk mengelola dan berbagi *database*. Data dapat disimpan dalam *file* sistem, dalam *database SQLite* atau dengan cara lain yang pada prinsipnya sama. *Content provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen

ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.2. PERT-CPM

Teknik PERT adalah suatu metode yang bertujuan untuk (semaksimal mungkin) mengurangi adanya penundaan kegiatan (proyek, produksi, teknik) maupun rintangan dan perbedaan-perbedaan; mengkoordinasikan dan menyelaraskan berbagai bagian sebagai suatu keseluruhan pekerjaan; dan mempercepat selesai proyek-proyek. Teknik ini merupakan suatu metode untuk menentukan jadwal dan anggaran dari sumber-sumber, sehingga suatu pekerjaan tertentu dapat diselesaikan tepat pada waktunya (Nurhayati, 2010).

Dengan teknik CPM penyusunan jaringan kerja diidentifikasi ke arah kegiatan serta menggunakan “*simple time estimates*” sebagai waktu pelaksanaan. Para pemakai teknik CPM dianggap mempunyai dasar yang lebih kuat sebagai landasan untuk memperkirakan waktu yang dibutuhkan untuk melaksanakan setiap kegiatan. Disamping itu di dalam proses perencanaan dan pengawasan dengan sistem ini turut diperhitungkan dan dimasukkan konsep biaya yang lebih mendetail sehingga memungkinkan pelaksanaan pembangunan proyek lebih singkat dan ekonomis (Nurhayati, 2010).

2.3. Jaringan Kerja

Analisa jaringan kerja proyek (*project network analysis*) adalah suatu sistem kontrol proyek yang berisi kegiatan tunggal, kegiatan gabungan, kegiatan paralel, dan lintasan kritis (Nurhayati, 2010). Analisa jaringan kerja berguna dalam mengkoordinir semua unsur proyek ke dalam suatu rencana utama (*master plan*) dengan menerapkan suatu metode kerja untuk melengkapi proyek sehingga diperoleh : (Nurhayati, 2010).

Pendekatan yang digunakan untuk pembentukan jaringan kerja adalah metode *Activity On Node* dan *Activity On Arrow*, kedua metode ini menggunakan dua bentuk blok-panah dan node. Metode pendekatan yang digunakan pada penelitian kali ini adalah metode AON (*Activity On Node*).

3. METODE PENELITIAN

3.1. Metode dan tool perancangan

USDP merupakan metode pengembangan/rekayasa perangkat lunak yang berbasis komponen (*component based software engineering*), yang berarti sistem perangkat lunak yang kelak dihasilkan akan terdiri atas komponen-komponen perangkat lunak yang saling terhubung melalui antarmuka yang terdefinisi dengan baik.

Dalam hal ini USDP, menggunakan UML sebagai kakas bantu utama analisis dan perancangan sistem perangkat lunak. USDP merupakan pengembangan sistem/perangkat lunak yang dikendalikan *use case*, sehingga *use case diagram* merupakan kendali dalam seluruh tahapan pengembangan sistem/perangkat lunak.

Unified Software Development Process (USDP), memiliki model-model sebagai berikut : 1) *Analysis*; 2) *Design*; 3) *Implementation*; 4) *Deployment*; 5) *Testing* (Nugroho, 2010).

UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Munawar, 2005). UML digunakan dalam pembuatan model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun. UML juga menggunakan class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek.

1. Diagram *Use Case*. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu.
2. Diagram *Sequence*. *Sequence diagram* menggambarkan interaksi antar obyek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (obyek-obyek yang terkait).
3. Diagram *Class*. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek.
4. Diagram *Activity*. Diagram *activity* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis, *Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana alir berakhir.

3.2. Metode dan konsep PERT-CPM-Network

Proses- proses yang digunakan dalam membangun jaringan kerja yaitu :

1. Kegiatan (*Activity*)
Untuk manajer proyek, suatu kegiatan merupakan elemen dari proyek yang membutuhkan waktu pelaksanaan (*duration*). Juga didefinisikan sebagai hal yang membutuhkan sejumlah sumber daya tenaga, aquipment, material, biaya dan sebagainya.
2. Kegiatan Memusat (*Merge Activity*)
Beberapa kegiatan yang berbeda lalu dilanjutkan dengan kegiatan yang sama sehingga disebut kegiatan memusat (lebih dari satu kaitan aliran panah)
3. Kegiatan Paralel (*Paralel Activity*)
Ini adalah kegiatan yang dikerjakan pada waktu yang bersamaan.
4. Alur (*Path*)
Alur merupakan suatu urutan koneksi, kegiatan yang terkait.
5. Alur kritis (*Critical Path*)
Ini berarti alur terpanjang yang terdapat pada jaringan. Jika terdapat suatu kegiatan yang tertunda (*delay*) pada alur, maka proyek juga akan tertunda pada waktu yang sama.
6. Kejadian (*Event*)
Istilah ini berupa suatu titik dan digunakan ketika sebuah kegiatan dimulai atau selesai. Jadi tidak membutuhkan waktu.
7. Kegiatan Memencar (*Burst Activity*)
Kegiatan ini memiliki lebih dari satu kegiatan yang secara bersamaan mengikutinya (lebih dari satu panah yang terkait mengikutinya).

Analisa jaringan kerja berguna dalam mengkoordinir semua unsur proyek ke dalam suatu rencana utama (*master plan*) dengan menerapkan suatu metode kerja untuk melengkapi proyek sehingga diperoleh :

1. Waktu terbaik untuk pelaksanaan (*best time*).
2. Pengurangan/penekanan ongkos/biaya (*least cost*).
3. Pengurangan resiko (*least risk*).
4. Mempelajari alternatif-alternatif yang terdapat di dalam dan di luar proyek.
5. Untuk mendapatkan atau mengembangkan *schedule* yang optimum.
6. Penggunaan sumber-sumber (*resources*) secara efektif dan efisien.
7. Alat komunikasi antar pimpinan.
8. Pengawasan pembangunan proyek.
9. Memudahkan revisi atau perbaikan terhadap penyimpangan yang terjadi

4. ANALISIS, PERANCANGAN DAN IMPLEMENTASI

4.1. Analisis Aplikasi Schedule Planning

Dasar aplikasi *schedule planning* yang dibangun disini adalah perencanaan jadwal dengan menggunakan metode manajemen produksi modern. Beberapa istilah yang dipergunakan di dalam perencanaan jadwal dengan metode manajemen produksi modern adalah :

1. PERT (*Product Evaluation Review Technique*)- merupakan metode mencari waktu paling optimal.
2. CPM (*Critical Path Method*) merupakan metode perhitungan untuk menentukan jalur kritis dimana sebuah list kerja tidak dapat ditunda.
3. Slack, merupakan perbedaan satuan waktu antara waktu paling lambat terwujudnya suatu list kerja dengan waktu paling cepat terjadinya list kerja.
4. Node, merupakan symbol untuk menentukan/menghubungkan antara list-list kerja dalam sebuah penjadwalan suatu proyek.

Metode perhitungan yang dilakukan terdiri atas 2 cara, yaitu cara perhitungan maju (*forward computation*) dan perhitungan mundur (*backward computation*), kemudian melakukan perhitungan *early start* (ES), *early finish* (EF), *latest start* (LS), *latest finish* (LF), setelah itu melakukan perhitungan *slack* untuk waktu luang dari setiap kegiatan. Untuk memperjelas perhitungan maju, perhitungan mundur dan *slack* dengan menggunakan gambar network jaringan kerja.

Perhitungan maju, perhitungangan bergerak dari awal *task*, menuju akhir *task*. Maksudnya ialah menghitung saat paling cepat terjadinya *task*, dan saat paling cepat dimulainya serta diselesaikannya *task*. Waktu pelaksanaan (durasi). Misalnya kegiatan A merupakan awal dari sebuah kegiatan sehingga saat tercepat diselesaikannya kegiatan A adalah pada hari kedua atau $EF_{(A)}$ adalah durasi dari kegiatan A.

Karena kegiatan B merupakan kegiatan yang dilaksanakan setelah kegiatan A maka:

$$EF_{(B)} = EF_{(A)} + Durasi_{(B)}$$

Misalnya kegiatan E merupakan kegiatan yang dilaksanakan setelah kegiatan C dan D maka kegiatan E akan dilaksanakan setelah kegiatan yang paling besar diantara kegiatan C dan D,

$$EF_{(C,E)} = EF_{(C)} + Durasi_{(E)}$$

$$EF_{(D,E)} = EF_{(D)} + Durasi_{(E)}, \text{ maka}$$

$$EF_{(E)} = \text{Max}(EF_{(C,E)}, EF_{(D,E)})$$

Perhitungan mundur, perhitungan dimulai dari akhir *task* menuju awal *task*, tujuannya ialah untuk menghitung saat paling lambat terjadinya *task* dan saat paling lambat dimulainya *task*.

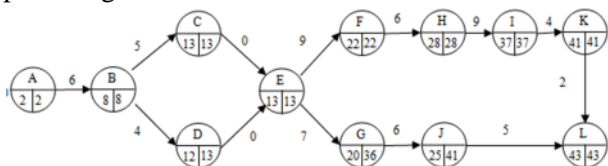
Misalnya dari hasil perhitungan maju diperoleh waktu maksimal selesai dari seluruh task adalah 43, sehingga dengan sendirinya .

Misalnya kegiatan C dan D merupakan kegiatan yang dilaksanakan setelah kegiatan E maka kegiatan C dan D akan dilaksanakan setelah kegiatan yang paling kecil diantara kegiatan E.

, maka

Setelah perhitungan maju dan perhitungan mundur selesai dilakukan, maka berikutnya dilakukan adalah perhitungan waktu luang/longgar dari setiap kegiatan atau *slack*. Perhitungan *slack* dilakukan dengan mencari selisih antara saat paling lambat diselesaikannya sebuah kegiatan dengan paling cepat diselesaikannya sebuah kegiatan. Misalnya kegiatan A maka perhitungan *slack* dari kegiatan A adalah

Berikut gambar contoh gambar diagram jaringan kerja perhitungan maju, perhitungan mundur dan perhitungan lintasan kritis *slack*



Gambar 4.1 Jaringan kerja perhitungan lintasan kritis

4.2. Analisis Masalah

Perencanaan jadwal sebagai salah satu aktifitas yang diabaikan, padahal dengan adanya perancaan ataupun manajemen jadwal yang baik dapat membuahkan hasil yang maksimal dan efisien. Tapi dalam implementasinya perencanaan jadwal biasanya dilakukan secara manual tanpa menggunakan metode kontrol yang baik sehingga hasil yang diperoleh tidak baik. Adapun yang telah menggunakan teknologi informasi belum digunakan secara maksimal, seiring dengan perkembangan teknologi *smartphone* yang semakin maju maka perencanaan jadwal dapat dilakukan pada *smartphone* untuk itu dibuatlah aplikasi *schedule planning* pada *smartphone* berbasis android yang memiliki fitur-fitur seperti *tasklist*, *agenda*, dan *memo*. *Schedule Planinng* juga memiliki keterbatasan karena hanya dapat melakukan perhitungan optimasi waktu, tidak

dapat menampilkan informasi visual dari diagram perencanaan jadwal dalam format grafik dan tidak dapat melakukan optimasi/percepatan biaya, pekerja, dan penentuan jadwal secara otomatis.

4.3. Analisis Kebutuhan

Secara umum berikut hasil analisis kebutuhan :

1. *User* dapat membuat perencanaan jadwal kerja secara terstruktur sehingga lebih efektif dan efisien.
2. *User* dapat mengidentifikasi task-task pekerjaan yang harus diselesaikan secepatnya pada proyek yang direncanakan.
3. *User* mengidentifikasi urutan jaringan pekerjaan node, sehingga berlangsung secara sistematis.
4. *User* dapat mengidentifikasi dan mengkalkulasi waktu perencanaan penyelesaian proyek berdasarkan node jaringan kerja dengan metode *critical path method*.
5. *User* dapat membuat agenda kerja/aktifitas yang dilakukan, dimana membantu user dalam manajemen kehidupan pribadi.
6. *User* dapat membuat catatan memo sebagai pengingat ataupun buku saku.

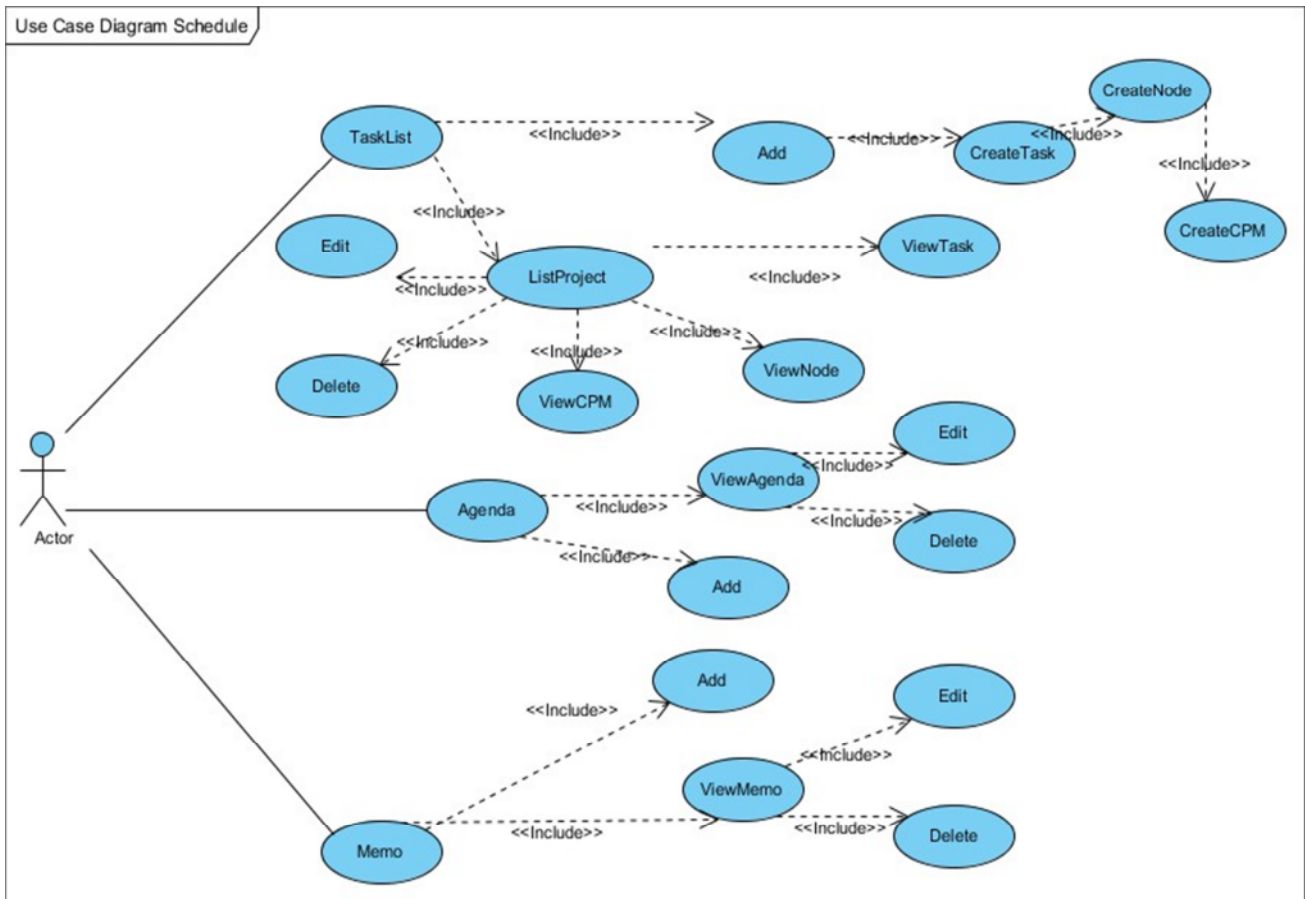
4.4. Solusi perancangan

Solusi dari keterbatasan yang telah dijelaskan permasalahan adalah dengan menghadirkan fitur-fitur yang dapat mengatasi kebutuhan tersebut pada aplikasi yang dibangun, sehingga mengatasi batasan yang dialami *user* saat ini.

4.5. Analisis kebutuhan fungsional

1. Use case Diagram

Dalam diagram use menyajikan interaksi antara *user* dan *use case*. Dimana pada aplikasi ini *user* merupakan aktor, dan *use case* merupakan representasi dari fungsional dari aplikasi. *User* dalam hal ini dapat menggunakan fungsi pada aplikasi ini yaitu tasklist, agenda, dan memo dalam manajemen jadwal user. Interaksi ini dapat dilihat dalam *Use case diagram* pada pada gambar 3.2 Di bawah ini.



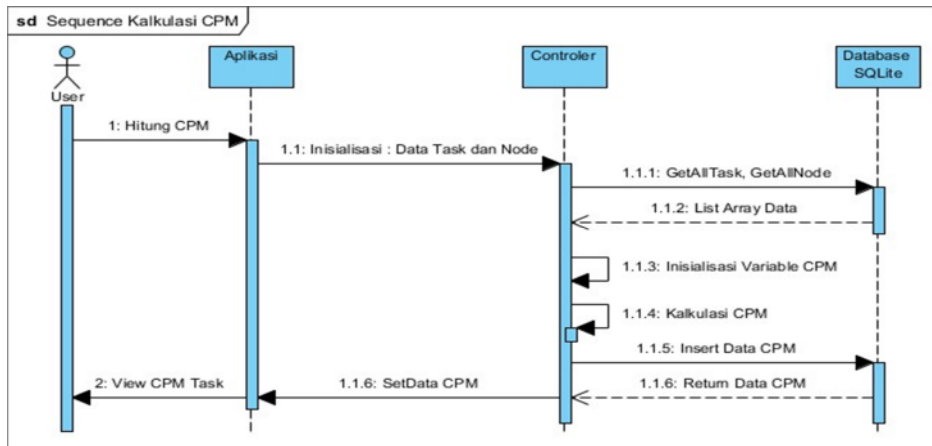
Gambar 4.2 Use case Diagram

2. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang digunakan untuk menggambarkan interaksi antar objek di dalam dan sekitar objek. Diagram sequence dapat dihasilkan dari setiap use case yang dianalisis.

Sequence Diagram Kalkulasi CPM

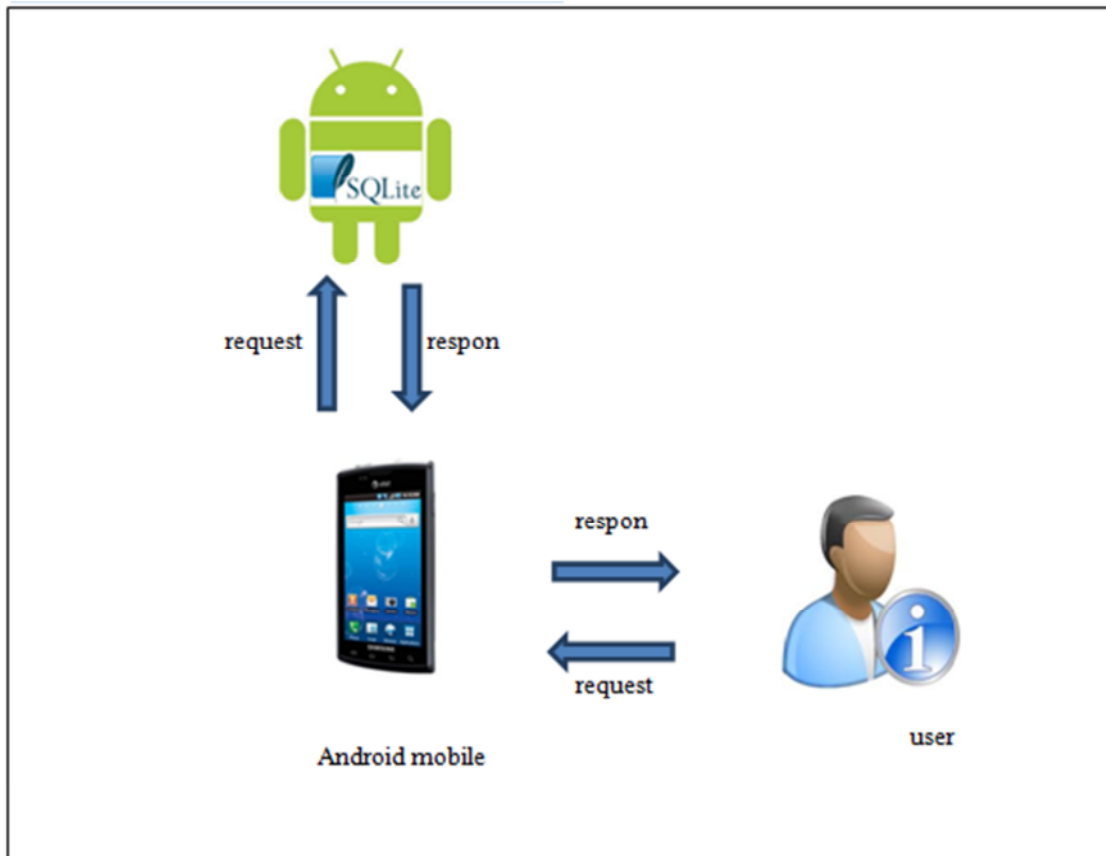
Setelah *User* telah menyelesaikan proses *create task* dan *create node*, user bisa melakukan kalkulasi CPM untuk menganalisa task mana saja yang tidak boleh waktu penyelesaiannya di tunda, setelah itu *user* menerima tampilan view CPM. *Sequence diagram* kalkulasi CPM dapat di lihat pada gambar di bawah ini.



Gambar 4.3 Sequence Diagram Kalkulasi CPM

4.6. Desain dan Perancangan Arsitektur

Arsitektur sistem aplikasi mikro blog twitter berbasis android dibuat untuk mengetahui bagaimana alur *user* mendapatkan informasi yang dibutuhkan. Arsitektur sistem dapat dilihat pada gambar berikut ini



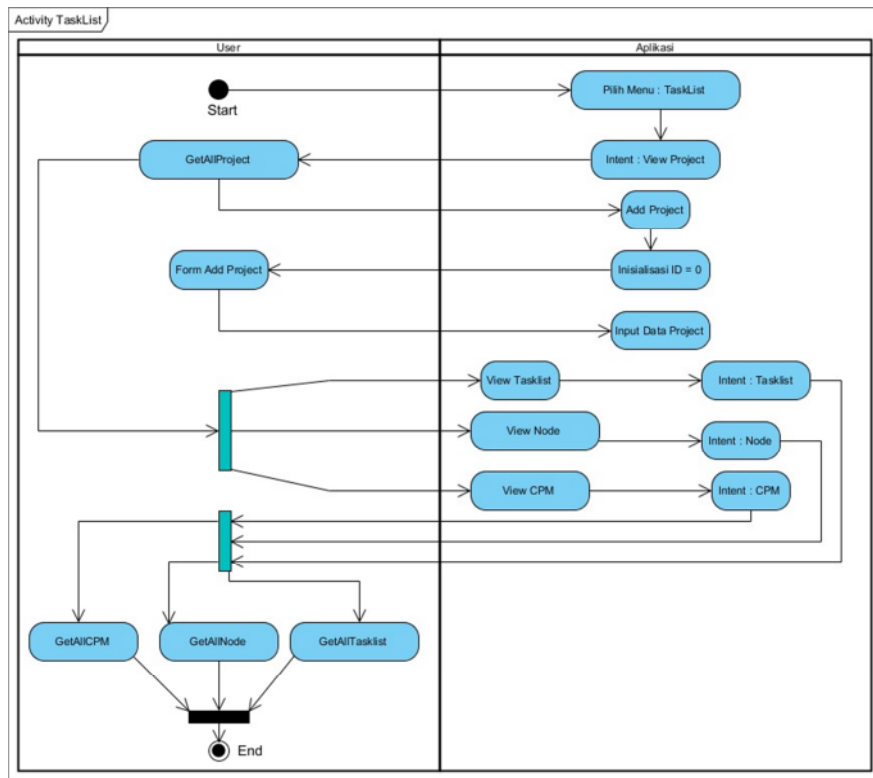
Gambar 4.4 Arsitektur Sistem *Schedule Planning*

1. Diagram Activity

Diagram Activity Tasklist

Diagram activity Tasklist memiliki dua aksi yaitu *user* dan aplikasi. Dalam sistem ini *user* telah melakukan pemilihan fitur menu tasklist pada aplikasi, system aplikasi akan menampilkan seluruh data project, kemudian *user* dapat melakukan request untuk menambah data project. Ketika *user* mengklik lama dari data project yang ditampilkan maka akan tampil menu contaxt yang berfungsi untuk melihat tasklist yang terdapat pada project itu, jaringan node kerja yang buat dan

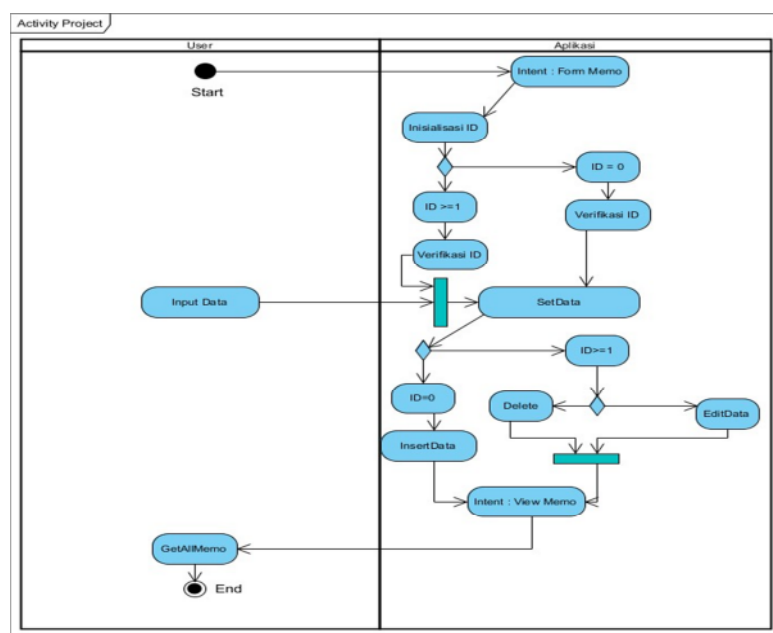
juga menampilkan hasil kalkulasi CPM. Setelah itu hasil atau *view* akan ditampilkan melalui ponsel yang dimiliki oleh *user*. *Activity Diagram Tasklist* dapat dilihat pada gambar berikut ini



Gambar 4.5 Activity Diagram Tasklist

Diagram Activity Project

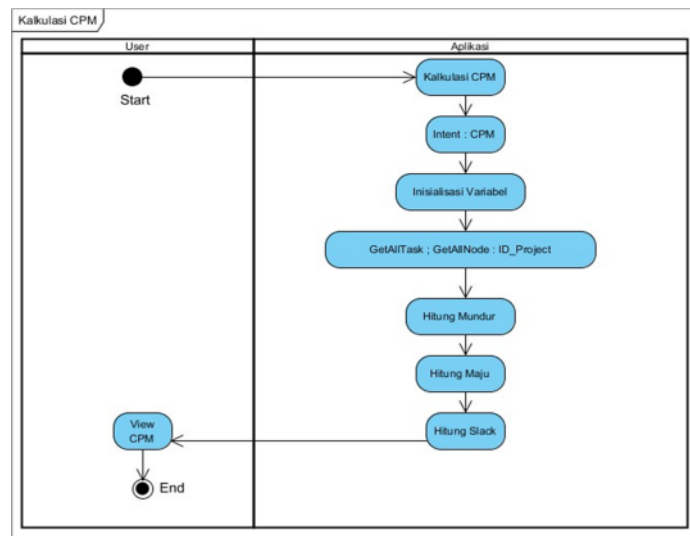
Diagram activity project memiliki dua aksi yaitu *user* dan aplikasi. Dalam sistem ini *user* memiliki akses untuk melakukan proses untuk add project akan dimulai dengan proses inisialisasi ID, verifikasi ID, dan semua prasyarat untuk melakukan input data project. Setelah itu hasil atau *view* akan ditampilkan melalui ponsel yang dimiliki oleh *user*. *Activity Diagram Project* dapat dilihat pada gambar berikut ini



Gambar 4.6 Diagram Activity Project

Diagram Activity CPM

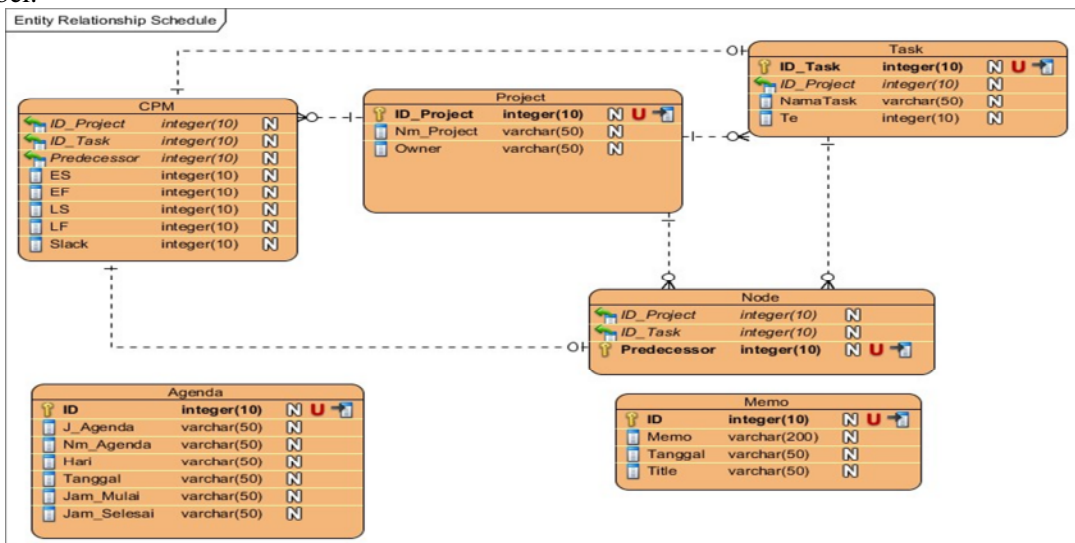
Diagram activity Kalkulasi CPM memiliki dua aksi yaitu *user* dan aplikasi. Dalam sistem ini *user* memiliki akses untuk melakukan pemilihan fitur kalkulasi CPM dimana fitur ini dapat diakses dengan prasyarat bahwa seluruh proses task telah dilakukan dan proses ini dapat dilakukan setelah proses terakhir selesai yaitu proses *create node*, dan setelah itu barulah melakukan proses kalkulasi CPM, *user* tidak melakukan input data text dimana sistem aplikasi melakukan proses inialisasi variabel CPM dan sistem melakukan request semua data yang berhubungan dengan *project*, kemudian sistem aplikasi melakukan activity untuk perhitungan mundur, maju dan *slack* dengan menggunakan variabel CPM, dari hasil perhitungan tersebut dapat ditentukan *task-task* yang mana saja mempengaruhi proses efisiensi dan efektifitas dari perencanaan penjadwalan. Setelah itu hasil atau *view* akan ditampilkan melalui ponsel yang dimiliki oleh *user*. *Activity Diagram Kalkulasi CPM* dapat dilihat pada gambar berikut ini.



Gambar 4.7 Diagram Activity CPM

2. Perancangan Database

Dalam pembuatan aplikasi *schedule planning* ini dengan membuat database dengan nama *schedule*, dengan tabel *project*, *task*, *node*, *agenda*, *memo* dan *cpm*, untuk tabel *project*, *task*, *node* dan *cpm* saling berelasi sedangkan untuk tabel *agenda*, dan *memo* tidak memiliki relasi. Berikut struktur tabel dan relasi antara tabel.



Gambar 4.8 Struktur dan relasi antar tabel

4.7. Implementasi

Perangkat keras yang digunakan dalam membangun aplikasi

Dalam pembuatan aplikasi schedule planning menggunakan platform android ini diperlukan perangkat keras untuk membantu dalam pembuatan aplikasi. Beberapa perangkat keras (*hardware*) yang digunakan saat pembuatan aplikasi dapat dilihat pada tabel berikut ini :

Tabel 4.1 Tabel kebutuhan perangkat keras (*hardware*)

No	Perangkat Keras	Keterangan
1	Proccesor	Intel ® B950 CPU 2.1 GHz
2	<i>Random Access Memory (RAM)</i>	DDR 2 Gb
3	<i>Video Graphics Accelerator (VGA)</i>	NVIDIA GeForce GT 520M
4	Hardisk	320 GB
5	Monitor	Asus LCD monitor 13.1
6	Handphone	LG Optimus 2x

Perangkat lunak yang digunakan dalam membangun aplikasi

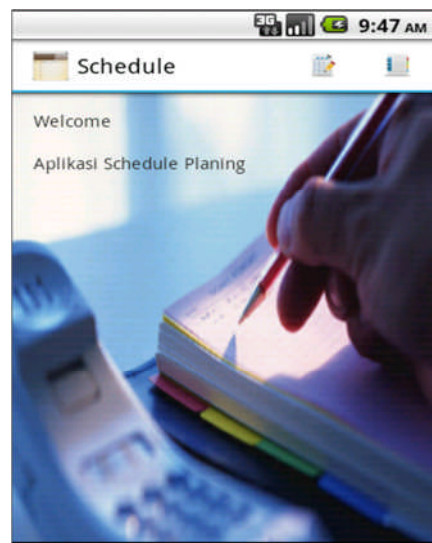
Dalam pembuatan aplikasi schedule planning menggunakan platform android ini diperlukan beberapa perangkat lunak untuk membantu pengerjaan aplikasi. Perangkat lunak (*software*) yang digunakan didalam pembuatan aplikasi ini dapat dilihat pada table berikut ini :

Tabel 4.2 Tabel kebutuhan perangkat lunak (*software*)

No	Perangkat Lunak	Keterangan
1	Windows seven	Sistem Operasi
2	Microsoft Office	Perangkat lunak untuk pembuatan laporan
3	<i>visual-paradigma</i>	Perangkat lunak untuk desain pemodelan UML
4	<i>Smartdraw</i>	Perangkat lunak untuk desain pemodelan Database
5	<i>Droiddraw</i>	Perangkat lunak untuk generate xml desain user interface
6	Eclipse IDE Juno	Perangkat lunak untuk code editor
7	Android SDK 2.2 (versi Froyo)	Perangkat lunak untuk debugging program dari code editor
8	Notepad ++	Perangkat lunak untuk code editor
9	SQLite	Perangkat lunak untuk pengolahan database

Dalam pembangunan aplikasi schedule planning menggunakan platform android ini memakai konsep pemrograman *model view control (MVC)*. Dimana pemrograman antara database, user interface dipisahkan dan dikontrol oleh controller.

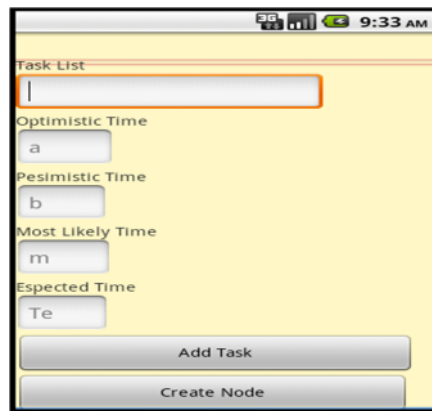
Halaman Welcome Screen



Gambar 4.9 Halaman Welcome Screen

Halaman Tasklist

Setelah melakukan pengiputan *project* baru maka akan tampil form input *tasklist* dengan menekan button *create task*. Halaman *form task* dapat dilihat pada gambar dibawah



Gambar 4.10 Form Task

Halaman CPM

Setelah melakukan pengiputan *node* baru maka akan tampil *form* CPM dengan menekan button Kalkulasi CPM. Halaman *form* CPM dapat dilihat pada gambar berikut :



Task Kegiatan	Te	ES	EF	LS
Analisis	4	0	4	0
Perancangan	5	4	9	-6
Desain UI	4	4	8	-10
Desain Database	5	9	14	0
Perograman	10	14	24	0
Implementasi	6	24	30	0

Gambar 4.11 Form CPM

Halaman Memo

Untuk melakukan input data untuk memo baru maka *user* bisa menekan tombol *button add* akan muncul dan tampil menu *form* memo. Halaman *form* memo dapat dilihat pada gambar berikut



Gambar 4.12 Form Memo

Halaman Agenda

Untuk melakukan input data untuk agenda baru maka *user* bisa menekan tombol menu maka akan muncul menu add dan akan tampil menu form agenda. Halaman *form* agenda dapat dilihat pada gambar dibawah



The image shows a mobile application interface for creating a new agenda item. The form is displayed on a yellow background and includes the following elements:

- A status bar at the top showing the time as 1:49 AM.
- A dropdown menu labeled "Pilih Kategori" (Select Category).
- A text input field labeled "Kegiatan" (Activity), which is highlighted with a red border.
- A dropdown menu labeled "Pilih Hari" (Select Day).
- A date input field showing "09/09/2014".
- A time input field labeled "Jam Mulai" (Start Time).
- A time input field labeled "Jam Selesai" (End Time).

Gambar 4.13 Form Agenda

5. PENUTUP

5.1 Kesimpulan

Dari hasil analisis dan perancangan telah berhasil dibuat sebuah aplikasi mobile schedule planning dengan menggunakan platform android. Aplikasi ini dapat mempermudah user dalam merencanakan sebuah *tasklist* dari sebuah *project* agar lebih efektif dan efisien. Selain itu user dapat membuat sebuah agenda rutin dan memo sehingga aktifitas user lebih tertata rapi.

5.2 Saran

Aplikasi mobile schedule planning menggunakan platform android dapat dikembangkan lagi dengan menggunakan optimasi biaya, optimasi resource, ataupun optimasi sumber daya manusia, sehingga sebuah proyek dan tasklistnya lebih terkontrol dengan baik. Selain itu aplikasi user juga dapat perubahan ataupun penambahan tasklist dan jaringan kerja (node), untuk segi visual kepada user diharapkan ada pengembangan tampilan userinterface agar lebih menarik dan juga terdapat visual seperti grafik ataupun diagram jaringan kerja. Selain itu untuk menambahkan fitur-fitur lain seperti setting ataupun pembatasan rule dalam penentuan jaringan kerja.

DAFTAR PUSTAKA

- A.S Rosa dan Shalahuddin M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Modula, Bandung.
- Dharwiyanti, Sri., dan Wahono, Romi Satrio, 2003. Pengantar Unified Modeling Language, Ilmukomputer.com, from <http://romisatriawahono.net>, dharwiyanti@rnd.inti.co.id.
- H, Safaat Nazruddin, 2012. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika Bandung, Bandung
- Hani T, Handoko, 1995. *Manajemen Personalial dan Sumber Daya Manusia*, BPFE-UGM, Yogyakarta.
- Haming, Murdifin dan Nurnajamuddin Mahfud, 2011. *Buku 1 Manajemen Produksi Modern Operasi Manufaktur dan Jasa Edisi Kedua*, Bumi Aksara, Jakarta.
- Hartono, Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*, Andi Offes, Yogyakarta.
- Kadir, Abdul, 2002. *Pengenalan Sistem Informasi*, Andi Offset, Yogyakarta.
- Kadir, Abdul, 2003. *Konsep dan Tuntunan Praktis Basis Data*, Andi Offset, Yogyakarta.
- Mulyadi, 2010. *Membuat Aplikasi untuk Android*, Multimedia Center Publishing, Yogyakarta.
- Nugroho Adi, 2010. *Rekaya Perangkat Lunak Berorientasi Objek dengan Metode USDP (Unified Software Development Process)*, Andi Offset, Yogyakarta.
- Nurhayati, 2010, *Manajemen Proyek*, Graha Ilmu, Yogyakarta.
- Sholih, 2006. *Pemodelan Sistem Informasi Berorientas Objek dengan UML*, Graha Ilmu, Yogyakarta.
- Siregar, Ivan, Yusuf, Ronald, Siendow, Welly, dan Wino, William, 2010. *Mengembangkan Aplikasi Enterprise berbasis Android*, Gava Media, Yogyakarta.
- Suyanto, 2010. *Algoritma Optimasi Deterministik atau Probabilitik*, Graha Ilmu, Yogyakarta.
- Wahana Komputer, 2013. *STEP by STEP menjadi Programer Android*, Andi Offset, Yogyakarta
- _____,
<http://elib.unikom.ac.id/files/disk1/300/jbptunikompp-gdl-adissyaefu-14989-3-babii_t-a.pdf>, (29 Juli 2012).
- _____,
<<http://zzzfadhlan.wordpress.com/2010/11/05/beberapa-pengertian-planning-perencanaan-dalam-organisasi/>>, (29 Juli 2012).
- _____,
<<http://anjarsaiangst.blogspot.com/2012/03/perkeembangan-android.html/>> (29 Juli 2012)