

**PERANCANGAN APLIKASI MOBILE REMOTE NETWORK
MONITORING MENGGUNAKAN MOBILE DEVICE BERBASIS
ANDROID**

SKRIPSI



**Disusun Oleh :
Dominikus Duli Kadu
Nim:09.12.504**



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015**

RESEARCH AND DEVELOPMENT
DEPARTMENT OF THE ARMY
WASHINGTON, D. C.

SECRET

Approved for Release
by NSA on 08-12-2013
 pursuant to E.O. 13526

THE NATIONAL SECURITY AGENCY
IS AUTHORIZED TO DISSEMINATE
THIS INFORMATION TO ANY
PERSON OR ORGANIZATION
WHICH IS DETERMINED TO BE
A "TRUSTED SOURCE" BY THE
DIRECTOR OF THE NATIONAL
SECURITY AGENCY

LEMBAR PERSETUJUAN

**PERANCANGAN APLIKASI MOBILE REMOTE NETWORK MONITORING
MENGUNAKAN MOBILE DEVICE BERBASIS ANDROID**

SKRIPSI

*Disusun Untuk Melengkapi Dan Memenuhi Persyaratan Guna Mencapai Gelar
Sarjana Teknik Elektro Sastra Satu (SI)*

Disusun oleh :

**Dominikus Duli Kadu
Nim : 09.12.504**

Diperiksa dan disetujui,

Pembimbing I

Pembimbing II

Dr. Eng. Arvanto Seotedjo, ST, MT
NIP.P. 1030800417

Bima Aulia Firmandani, ST

Mengetahui,



M. H. Anim Ashari, ST, MT
NIP.P.1030100358

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER**

2015



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

PROGRAM PASCASARJANA MAGISTER TEKNIK

PT.BNI(PERSERO)MALANG Kampus I :Jl. Bendungan Sigura-gura No.2 Telp. (0341)551431(Hunting), Fax (0341)553015 Malang 65145
BANK NIAGA MALANG Kampus II :Jl.Raya Karanglo, Km.2 Telp.(0341)417636 Fax (0341)417634 malang

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini :

Nama : Dominikus Duli Kadu
NIM : 09.12.504
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang....03-2015

g membuat pernyataan,



Dominikus Duli Kadu

09.12.504

PERSEMBAHAN

Pujih dan syukur penulis panjatkan kehadirat Tuhan Yang Esa atas segala limpahan rahmad, yang diberikan-Nya kepada penulis sehingga penulisan skripsi ini dapat terselesaikan.

Hari takkan indah tanpa mentari dan rembulan, begitu juga hidup takkan indah tanpa tujuan, harapan serta tantangan. Meski terasa berat, namun manisnya hidup justru akan terasa, apabila semuanya terlalui dengan baik, meski harus memerlukan pengorbanan.

Dari semua tlah Kau tetapkan hidupku dalam tangan-Mu, dalam takdir-Mu, rencana indah yang tlah Kau siapkan bagi masa depanku yang penuh harapan, harapan kesuksesan terpangku di pundak sebagai janji kepada mereka IBU dan BAPAK.

Untuk yang pertama Ku persembahkan Skripsi ini kepada Orang Tua Ku Ayahanda Yohanes Liat dan Ibunda Lusia Lou. Sosok yang pertama dari tujuan hidupku yang selalu membangkitkan dalam keterpurukan ku. Terimakasih ya Tuhan yang memberikan malaikat-Mu kepada Ku.

Kakakku Rosa Dalimia Wae Raya, Veronika Tuto dan Nadus Riantobi yang ku sayang, yang selalu mendukung dan mengingatkanku untuk selalu semangat dalam mengerjakan skripsiku ini.

Seluruh keluarga besar saya di rumah yang telah memberikan dukungan, perhatian, motivasidan do'a hingga terselesaikannya penulisan skripsi ini.

Nila Wahyu Maya yang selalu setia menemani, memberi dukungan, dan membantu dikala kesulitan dalam penyelesaian skripsi ini.

Teman-teman kost 128 (Yanceko, Eman, Kader, Ardrian, Cecep dan Danar, Astin opu, Yanus) terimakasih atas dukungan dan kebersamaannya.

Semoga Tuhan senantiasa menunjukkan jalan dan memberikan cahaya-Nya, serta melampangkan dada kita dengan limpahkan iman dan keindahan tawakal kepada-Nya.

Salam Elektro S-1

Amdoel tobi

PERANCANGAN APLIKASI MOBILE REMOTE NETWORK MONITORING MENGGUNAKAN MOBILE DEVICE BERBASIS ANDROID

Dominikus Duli Kadu

09.12.504

Jurusan Teknik Elektro, Konsentrasi Teknik Komputer

Falkutas Teknologi Industri

Institut Teknologi Nasional Malang

Jln.Raya Karanglo Km 2 Malang

Email : Tobi844@yahoo.co.id

Abstrak

Smartphone merupakan alat komunikasi yang tidak menggunakan kabel dan dapat dibawa kemana-mana. Pada umumnya ponsel yang menggunakan system operasi Android mempunyai fungsi lebih luas serta berbagai aplikasinya memberikan kemudahan bagi kita dalam melakukan berbagai aktifitas sehari-hari, seperti di bidang kontrol, di bidang entertainment, dan di bidang komunikasi yang menyebabkan berubahnya pola hidup di dalam masyarakat menjadi sesuatu yang serba cepat, tepat, dan efisien. Sistem operasi ini bersifat open source sehingga terdapat banyak ruang untuk mengembangkan dan membangun berbagai macam aplikasi sesuai dengan kebutuhan yang diinginkan.

Dengan keleluasaan ini maka muncul gagasan untuk melakukan penelitian dalam membuat Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android yang berfungsi untuk membantu administrator lebih efektif mengawasi dan memonitoring jaringan.

Kata Kunci : *Simtem Monitoring Jaringan*

KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya sehingga penelitian berjudul **Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android**.

Penelitian ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terimakasih yang sebesar-besarnya kami sampaikan pada:

1. Dr. Ir Lalu Mulyadi, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Dr. Eng. Aryoanto Soetedjo, ST, MT selaku Dosen Pembimbing satu Tugas Skripsi.
5. Bima Aulia Firmandani, ST selaku Dosen Pembimbing dua Tugas Skripsi.
6. Sahabat-sahabat dan rekan-rekan yang tidak kami sebutkan satu-persatu, kami ucapkan banyak terimakasih atas bantuannya dalam proses pembuatan Skripsi yang telah saya kerjakan, begitu juga dengan penyelesaian laporan ini.

Usaha ini telah kami lakukan semaksimal mungkin, namun jika ada kekurangan dan kegagalan dalam penyusunan, kami mohon saran dan kritik yang sifatnya membangun. Begitu juga sangat kami perlukan untuk menambah kesempurnaan laporan ini dan dapat bermanfaat bagi rekan-rekan mahasiswa pada khususnya dan pembaca pada umumnya.

Malang, Februari 2015

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	2
1.5. Metodologi Penelitian	2
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	4
2.1. Jaringan Komputer.....	4
2.1.1 Jenis Jaringan Komputer	5
2.1.2 Pengertian Monitoring Jaringan Komputer.....	6
2.2.3 Tujuan Monitoring Jaringan Komputer	7
2.2.4 Tahapan Monitoring Jaringan Komputer	7
2.2.5 Proses Di Dalam System Monitoring.....	8
2.2.6 Protokol Jaringan Komputer	9
2.2.7 Topologi Jaringan Komputer	9
2.2 Open Systems Interconnection (OSI).....	15
2.3 Simple Network Management Protocol (SNMP).....	19
2.3.1 Management Information Base (MIB).....	19
2.3.2 Arsitektur SNMP	21
2.3.3 Protokol SNMP	22
2.3.4 Perkembangan dan penggunaan SNMP	23
2.3.5 Mengimplementasikan SNMP	24

2.4 Perkembangan Android	24
2.5 Software Development Kit (SDK) Eclipse	28
2.5.1 Sejarah	29
2.5.2 Arsitektur	29
2.5.3 Histori (versi-versi) Eclipse	30
BAB III DESAIN DAN IMPLEMENTASI SISTEM	31
3.1. Deskripsi Sistem	31
3.2 Analisa sistem	32
3.2.1 Analisa Masalah	32
3.2.2 Analisa Data	32
3.2.3 Analisa Perangkat Lunak	34
3.2.4 Analisa Perangkat Keras	34
3.3.5 Analisa User	35
3.3 Tahap Perancangan	35
3.3.1 Rancangan Use Case Diagram	35
3.3.2 Rancangan Diagram Alir	36
3.3.3 Desain Atau Interface Tampilan Awal	37
3.3.4 Desain Atau Interface Tampilan Hasil	37
3.3.5 Perancangan Android <i>Project</i> Pada <i>Eclipse</i>	37
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	51
4.1. Implementasi Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android	51
4.2. Pengujian Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android	51
4.3. Tampilan Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android	51
4.3.1 Icon Aplikasi Pada Ponsel.	52
4.3.2 Tampilan Awal Aplikasi..	53
4.3.3 Tampilan Menu Input Ip Jaringan..	53

4.3.4 Tampilan Monitoring.....	54
4.3.5 Tampilan Hasil Monitoring Devices.....	55
4.3.6 Tampilan Hasil Monitoring Traffic Jaringan.....	56
4.3.7 Tampilan Hasil Monitoring Bandwidth Jaringan.....	57
4.3.8 Tampilan Hasil Monitoring Ip Address List.....	59
4.4 Tabel Hasil Pengujian.....	60
4.4.1 Penujian Hari Pertama.....	60
4.4.2 Penujian Hari Kedua.....	64
4.4.3 Pengujian Hasil Monitoring Jaringan Menggunakan Aplikasi dan Pc.....	67
BAB V PENUTUP.....	70
5.1. Kesimpulan.....	70
5.2. Saran.....	70
DAFTAR PUSTAKA.....	71

DAFTAR GAMBAR

2.1. Jaringan Komputer	4
2.2. Proses Dalam Sistem Monitoring Jaringan Komputer	7
2.3. Topologi BUS	10
2.4. Topologi Ring	11
2.5. Topologi Star	12
2.6. Topologi Tree	13
2.7. Topologi Mesh	14
2.8. Topologi Linear	15
2.9. Lapisan Pada Osi Layer	16
2.10. Struktur SNMP	19
2.11. Objek Identifier Dalam MIB	21
2.12. Logo Android	25
2.13. Logo Software Development Kit Eclipse	28
3.1. Use Case Diagram	34
3.2. Rancangan Diagram Alir	35
3.3. Desain atau Interface Tampilan Awal	36
3.4. Desain Atau Interface Tampilan Hasil	37
3.5. Tampilan Eclipse	38
3.6. Tampilan Android Project	38
3.7. Tampilan Nama Project	38
3.8. Tampilan Select Buid Target	39
3.9. Tampilan Application Info	39
3.10. Home Main Activity.java	40
3.11. Tampilan <i>Splash Screen</i>	41
3.12. Tampilan Awal	43
3.13. Tampilan Hasil	48
3.14. Tampilan menu Run As Application	49
3.15. Tampilan Menu Emulator	50

4.1. Tampilan Icon Aplikasi.....	52
4.2. Tampilan Awal aplikasi	53
4.3. Tampilan Menu Input IP Jaringan.....	53
4.4. Tampilan Monitoring	54
4.5. Tampilan Hasil Monitoring Devices	55
4.6. Tampilan Hasil Monitoring Devices	55
4.7. Tampilan Hasil Monitoring Traffic Jaringan	56
4.8. Tampilan Hasil Monitoring Traffic Jaringan	56
4.9. Tampilan Hasil Monitoring Bandwidth Jaringan.....	57
4.10. Tampilan Hasil Monitoring Bandwidth Jaringan.....	58
4.11. Tampilan Hasil Monitoring Ip Address List	59
4.12. Tampilan Hasil Monitoring Ip Address List	59

DAFTAR TABEL

Tabel 2.1. Protokol data unit (PDU) dari SNMP	22
Tabel 4.1 Hasil Monitoring Device Pada Hari Pertama.....	60
Tabel 4.2 Hasil Monitoring Traffic Jaringan Pada Hari Pertama.....	61
Tabel 4.3 Hasil Monitoring Bandwidth Jaringan Pada Hari Pertama	62
Tabel 4.4 Hasil Monitoring Ip Address List	62
Tabel 4.5 Hasil Monitoring Device Pada Hari Kedua.....	64
Tabel 4.6 Hasil Monitoring Traffic Jaringan Pada Hari Kedua	65
Tabel 4.7 Hasil Monitoring Bandwidth Jaringan Pada Hari Kedua.....	66
Tabel 4.8 Hasil Monitoring Ip Address List Pada Hari Kedua	66
Tabel 4.9 Hasil Monitoring Traffic Jaringan Menggunakan Aplikasi Skripsi dan Pc.....	68
Tabel 4.10 Hasil Monitoring Bandwidth Jaringan Menggunakan Aplikasi Skripsi dan Pc	68
Tabel 4.11 Hasil Monitoring Ip Address Lis Menggunakan Aplikasi Skripsi dan Pc.....	69

BAB I

PENDAHULUAN

1.1. Latar Belakang

Salah satu tugas dasar dari administrator jaringan adalah pemeliharaan jaringan, hal ini dapat dilakukan dengan cara memantau jaringan yang dikelola tersebut secara berkala. Sistem pemantauan atau monitoring jaringan kini dapat dilakukan dari jarak jauh, sehingga jaringan dapat dipantau kinerjanya setiap saat.

Teknologi mobile device yang ada saat ini memungkinkan pengguna untuk dapat terhubung dengan internet dan bertukar data. Selain itu, mobile device yang ada saat ini sudah mendukung penggunaan system operasi. Salah satu system operasi mobile device yang banyak digunakan pada telepon seluler adalah system operasi android, Sistem operasi ini bersifat open source sehingga terdapat banyak ruang untuk mengembangkan dan membangun berbagai macam aplikasi sesuai dengan kebutuhan yang diinginkan.

Dengan keleluasaan ini maka muncul gagasan untuk membangun sebuah sistem monitoring jaringan yang dapat di jalankan pada mobile device. Dengan aplikasi monitoring jaringan yang terdapat pada mobile device ini maka lebih mudah bagi administrator jaringan untuk memonitoring jaringan yang dikelolanya secara mobile menggunakan mobile device berbasis Android, kapanpun administrator jaringan ingin mengetahui kondisi jaringan yang dikelolanya. Hal inientunya menjadi keunggulan tersendiri bagi administrator jaringan untuk dapat memonitoring jaringan yang dikelolanya secara mobile, dimana hal ini tidak dapat dimungkinkan apabila menggunakan PC. Administrator tinggal menggunakan aplikasi ini untuk mengetahui kondisi jaringan yang di kelolanya. Berdasarkan latar belakang tersebut maka penulis membuat skripsi dengan judul **PERANCANGAN APLIKASI MOBILE REMOTE NETWORK MONITORING MENGGUNAKAN MOBILE DEVICE BERBASIS ANDROID.**

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka masalah yang dapat dibahas adalah mengenai bagaimana cara membuat perancangan aplikasi mobile remote network monitoring menggunakan mobile device berbasis android

1.3. Tujuan Penelitian

Tujuan pembuatan program ini ialah aplikasi ini di buat untuk membantu administrator jaringan agar dapat mengakses dan memonitor jaringan yang dikelolanya secara mobile, kapanpun administrator jaringan ingin mengetahui kondisi jaringan yang dikelolanya.

1.3. Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Aplikasi ini di uji di sistem operasi android minimal versi.2,3 gingerbread.
2. Software yang digunakan adalah Eclipse sebagai IDE JAVA, android SDK sebagai Emulator Android.
3. Simple Network Management Protocol (SNMP) yang di gunakan tiga layer yaitu physical, data link, network.

1.5. Metodologi Peneliti

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studiliteratur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa kebutuhan aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar dihasilkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan aplikasi dimana nantinya akan digunakan sebagai acuan perancangan aplikasi.

3. Perancangan dan implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa

kebutuhan untuk membangun aplikasi ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari aplikasi yang akan dibuat dan di implementasikan kedalam aplikasi.

4. Eksperimen dan evaluasi

Pada tahap ini, aplikasi yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut:

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

Bab III : Perancangan dan Analisa sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

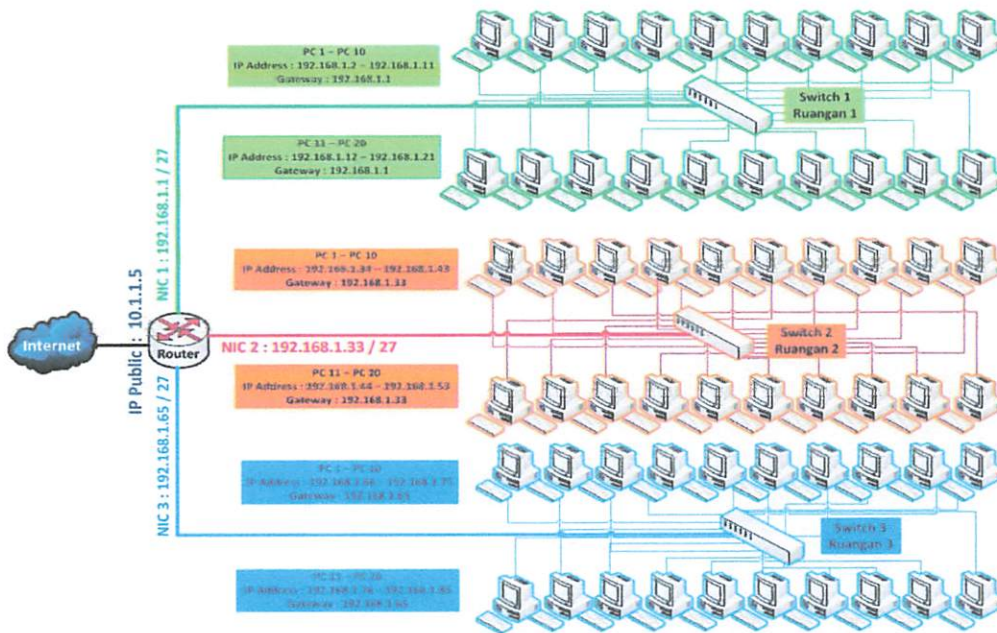
Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan Komputer adalah sekelompok komputer yang saling berhubungan anantara satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi seperti kabel, gelombang radio, saluran telepon, satelit, maupun serat optik sehingga dapat saling berinformasi, program-program pengguna bersama perangkat keras seperti printer, harddisk, dan lain sebagainya. Atau jaringan komputer bisa di artikan sebagai kumpulan sejumlah terminal komunikasi yang berbeda di berbagai lokasi yang terdiri dari lebih atau satu komputer yang saling berhubungan.



Gambar 2.1 Jaringan Komputer [1]

Jaringan komputer merupakan suatu sistem yang terdiri komputer dan perangkat lainnya yang dirancang untuk dapat bekerja bersama-sama dalam berbagai manfaat dan tujuan antara lain untuk berkomunikasi, akses informasi, menerima maupun memberikan layanan. Bagian yang menerima layanan disebut *Client* dan bagian yang memberikan layanan disebut *Server*.

2.1.1 Jenis Jaringan Komputer

Secara umum jaringan komputer dibagi atas lima jenis, yaitu;

1. Local Area Network (LAN)

Local Area Network (LAN), merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumber daya (misalnya printer) dan saling bertukar informasi.

2. Area Network (MAN)

Metropolitan Area Network (MAN), pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum.

3. Wide Area Network (WAN)

Wide Area Network (WAN), jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin-mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai.

4. Internet

Orang yang terhubung ke jaringan sering berharap untuk bisa berkomunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antar jaringan yang seringkali tidak kompatibel dan berbeda.

5. Jaringan Tanpa Kabel

Jaringan tanpa kabel merupakan suatu solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang,

maka mutlak jaringan tanpa kabel diperlukan karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat. Saat ini jaringan tanpa kabel sudah marak digunakan dengan memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang sistem menggunakan kabel.

2.1.2 Pengertian Monitoring Jaringan Komputer

Kata *network monitoring* dapat menggambarkan bagaimana suatu secara konstan memonitor kondisi jaringan seperti *router*, *server* maupun *device* lainnya. Apabila pada suatu sistem jaringan terjadi suatu *error* atau *down* maka secara otomatis keadaan tersebut akan diketahui *network administrator* atau *system administrator* melalui sebuah sistem peringatan berupa *email* ataupun notifikasi lainnya. Maka dalam hal ini, seorang *network administrator* memiliki andil dalam memonitor atau memantau jaringan agar jaringan selalu berada dalam kondisi optimal. Monitoring jaringan komputer dapat berupa suatu koleksi informasi (data) yang merupakan salah satu fungsi dari manajemen jaringan yang berguna untuk menganalisa apakah suatu jaringan masih cukup layak untuk digunakan untuk berkomunikasi atau memerlukan tambahan kapasitas. Hasil dari monitoring dapat membantu jika admin ingin mendesain ulang jaringan yang telah ada. Banyak hal dalam jaringan yang biasa dimonitoring, salah satu diantaranya yaitu *load traffic* jaringan yang lewat pada sebuah *router* atau *interface* komputer.

Monitoring komputer di bagi menjadi dua bagian:

1. *Connection monitoring*

Connection monitoring adalah teknik monitoring jaringan yang dapat dilakukan dengan melakukan tes ping antara *monitoring station* dan *device target*, sehingga dapat diketahui bila koneksi terputus.

2. *Traffic Monitoring*

Traffic monitoring adalah teknik monitoring jaringan dengan melihat paket aktual dari *traffic* pada jaringan dan menghasilkan laporan berdasarkan *traffic* jaringan.

2.1.3 Tujuan Monitoring Jaringan Komputer

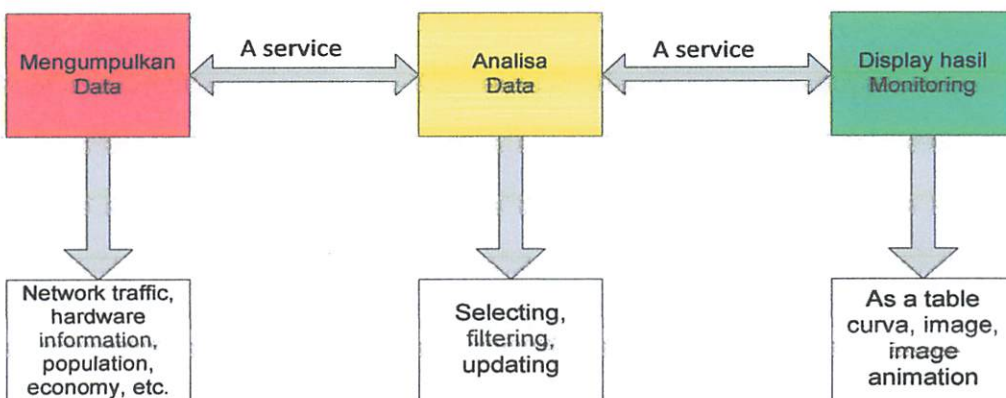
Tujuan monitoring jaringan komputer adalah untuk mengumpulkan informasi yang berguna dari berbagai bagian jaringan sehingga jaringan dapat diatur dan dikontrol dengan menggunakan informasi yang telah terkumpul. Dengan begitu diharapkan jika terjadi *trouble* atau permasalahan dalam jaringan akan cepat diketahui dan diperbaiki sehingga stabilitas jaringan lebih terjamin. Berikut ini beberapa alasan utama dilakukan monitoring jaringan :

- Untuk menjaga stabilitas jaringan.
- Sulit untuk mengawasi apa yang sedang terjadi di dalam jaringan yang memiliki sejumlah besar mesin (*host*) tanpa alat pengawas yang baik.
- Untuk mendeteksi kesalahan pada jaringan, *gateway*, *server*, maupun *user*.
- Untuk memberitahu *trouble* kepada administrator jaringan secepatnya. Mempermudah analisis *troubleshooting* pada jaringan.
- Mendokumentasikan jaringan.

2.1.4. Tahapan Monitoring Jaringan Komputer

Secara garis besar tahapan dalam sebuah sistem monitoring terbagi ke dalam tiga proses besar, yaitu:

1. Proses di dalam pengumpulan data monitoring.
2. Proses di dalam analisis data monitoring.
3. Proses di dalam menampilkan data hasil monitoring.



Gambar 2.2 Proses Dalam Sistem Monitoring Jaringan Komputer [2]

Sumber data dapat berupa *Bandwidth Jaringan*, informasi mengenai *hardware*, atau sumber-sumber lain yang ingin diperoleh informasi mengenai dirinya. Proses dalam analisis data dapat berupa pemilihan data dari sejumlah data telah terkumpul atau bisa juga berupa manipulasi data sehingga diperoleh informasi yang diharapkan. Sedangkan tahap menampilkan data hasil monitoring menjadi informasi yang berguna di dalam pengambilan keputusan atau kebijakan terhadap sistem yang sedang berjalan dapat berupa sebuah tabel, gambar, gambar kurva, atau dapat juga berupa gambar animasi. Aksi yang terjadi di antara proses-proses yang ada di dalam sebuah sistem monitoring adalah berbentuk *service*, yaitu suatu proses yang terus-menerus berjalan pada interval waktu tertentu. Proses yang dijalankan dapat berupa pengumpulan data dari objek yang dimonitor, atau melakukan analisis data yang telah diperoleh dan menampilkannya. Proses yang terjadi tersebut bisa saja memiliki interval waktu yang berbeda. Pada dasarnya banyak sekali *software* baik itu *freeware* maupun berbayar yang digunakan untuk melakukan monitoring jaringan.

2.1.5 Proses Di Dalam Sistem Monitoring

Sumber data dapat berupa *network traffic*, informasi mengenai *hardware*, dan lain sebagainya. Proses dalam analisis data dapat berupa pemilihan data dari sejumlah data yang telah terkumpul atau bisa juga berupa manipulasi data sehingga diperoleh informasi yang diharapkan. Sedangkan tahap menampilkan data hasil monitoring menjadi informasi yang berguna di dalam pengambilan keputusan atau kebijakan terhadap sistem yang sedang berjalan dapat berupa sebuah tabel, gambar, kurva, atau animasi. Aksi yang terjadi diantara proses-proses yang ada di dalam sebuah sistem monitoring adalah berbentuk *service*, yaitu suatu proses yang terus-menerus berjalan pada interval waktu tertentu. Proses yang dijalankan dapat berupa pengumpulan data dari objek yang dimonitor atau melakukan analisis data yang telah diperoleh dan menampilkannya. Proses yang terjadi tersebut bisa saja memiliki interval waktu yang berbeda. Contoh interval waktu didalam pengumpulan data dapat terjadi tiap lima menit sekali. Namun pada proses analisis data terjadi tiap satu jam sekali untuk menghasilkan informasi yang diharapkan membutuhkan lebih dari satu sampel data, misal untuk nilai rata-rata data (*average*) dengan sebanyak 60 sampel data.

2.1.6 Protokol Jaringan Komputer

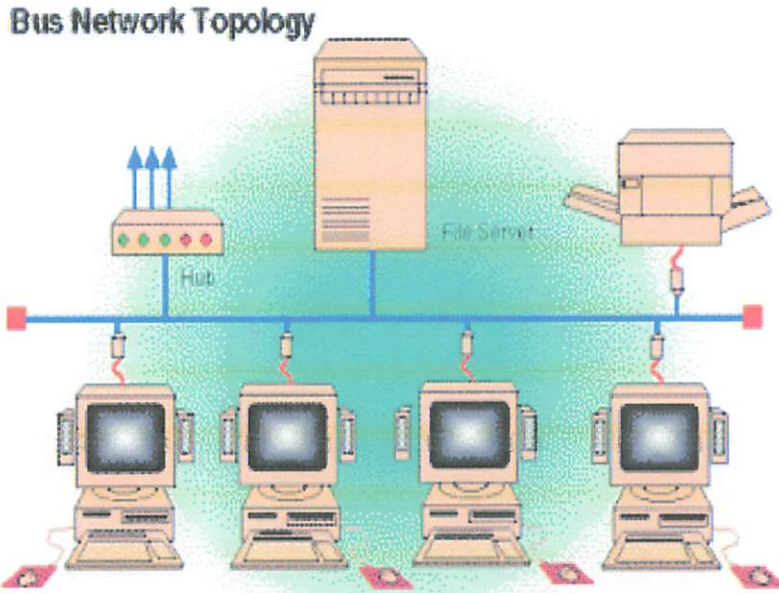
Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras. Protokol digunakan untuk menentukan jenis layanan yang akan dilakukan pada internet: *TCP/IP (singkatan dari Transmission Control Protocol/Internet Protocol)*, *UDP (User Datagram Protokol)*, *Domain Name System (DNS)*, *Point-to-Point Protocol*, *Serial Line Internet Protocol*, *Internet Control Message Protocol (ICMP)*, *POP3 (Post Office Protocol)*, *IMAP (Internet Message Access Protocol)*, *SMTP (Simple Mail Transfer Protocol)*, *HTTP (Hypertext Transfer Protocol)*, *HTTPS*, *SSH (Sucure Shell)*, *Telnet (Telecommunication network)*, *FTP (File Transfer Protocol)*, *LDAP*, *SSL (Secure Socket Layer)*.

2.1.7 Topologi Jaringan Komputer

Topologi jaringan adalah suatu aturan atau cara untuk menghubungkan komputer yang satu dengan komputer yang lainnya sehingga membentuk suatu jaringan. Topologi jaringan juga dapat didefinisikan sebagai gambaran secara fisik dari pola hubungan antara komponen jaringan, yang meliputi *Server*, *Workstation*, *Hub*, dan pengkabelannya. Macam-macam Topologi Jaringan Komputer :

1. Topologi BUS

Jenis topologi bus ini menggunakan kabel tunggal, seluruh komputer saling berhubungan secara langsung hanya menggunakan satu kabel saja. Kabel yang menghubungkan jaringan ini adalah kabel koaksial dan dilekatkan menggunakan *T-Connector*. Untuk memaksimalkan penggunaan jaringan ini sebaiknya menggunakan kabel *Fiber Optic* karena kestabilan resistensi sehingga dapat mengirimkan data lebih baik



Gambar 2.3 Topologi BUS [2].

Kelebihan Topologi Bus

1. Mudah untuk dikembangkan
2. Tidak memerlukan kabel yang banyak
3. Hemat biaya pemasangan.

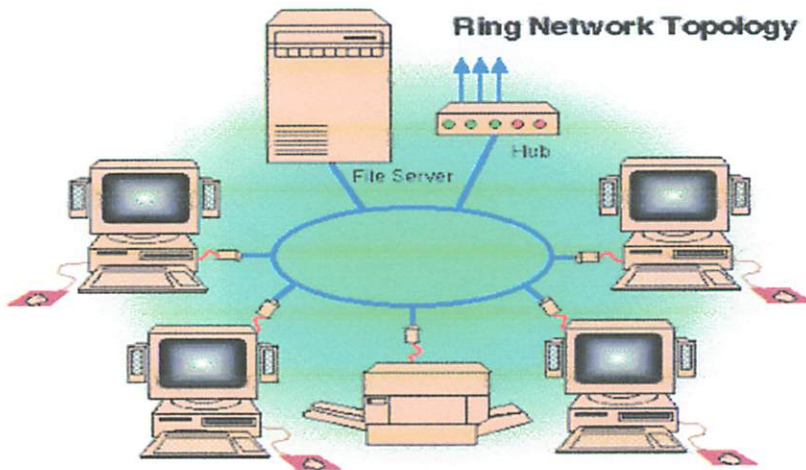
Kelemahan topologi bus :

1. Tidak stabil, jika salah satu komputer terganggu maka jaringan akan terganggu
2. Tingkat deteksi kesalahan sangat kecil
3. Sulit mencari gangguan pada jaringan
4. Tingkat lalu lintas tinggi / sering terjadi antrian data
5. Untuk jarak jauh diperlukan repeater

2. Topologi Ring

Jenis topologi ring ini, seluruh komputer dihubungkan menjadi satu membentuk lingkaran (ring) yang tertutup dan dibantu oleh Token, Token berisi informasi yang berasal dari komputer sumber yang akan memeriksa apakah informasi tersebut digunakan oleh titik yang bersangkutan, jika ada maka token akan memberikan data yang diminta oleh titik jaringan dan menuju ke titik berikutnya. seluruh komputer akan menerima setiap signal informasi yang

mengalir, informasi akan diterima jika memang sudah sesuai dengan alamat yang dituju, dan signal informasi akan diabaikan jika bukan merupakan alamatnya sendiri. Dengan kata lain proses ini akan berlanjut terus hingga sinyal data diterima ditujuan.



Gambar 2.4 Topologi Ring [2]

Kelebihan :

1. Tidak menggunakan banyak kabel
2. Tingkat kerumitan pemasangan rendah
3. Mudah instalasi
4. Tidak akan terjadi tabrak data
5. Mudah dirancang

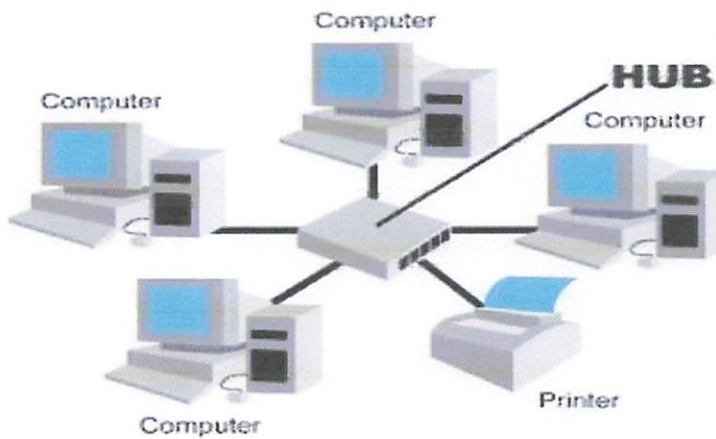
Kekurangan :

1. peka kesalahan jaringan
2. Sulit untuk dikembangkan
3. Jika salah satu titik jaringan terganggu maka seluruh komunikasi data dapat terganggu

3. Topologi Star

Pada topologi jenis star ini, setiap komputer langsung dihubungkan menggunakan Hub, dimana fungsi dari Hub ini adalah sebagai pengatur lalu lintas seluruh komputer yang terhubung. Karena menggunakan proses

pengiriman dan penerimaan informasi secara langsung inilah yang menyebabkan biaya pemasangannya juga tinggi.



Gambar 2.5 Topologi Star [2]

Kelebihan :

1. Deteksi kesalahan mudah dilakukan
2. Perubahan stasiun mudah dilakukan dan tidak mengganggu jaringan lain
3. Mudah melakukan control
4. Tingkat keamanan tinggi
5. Paling fleksibel

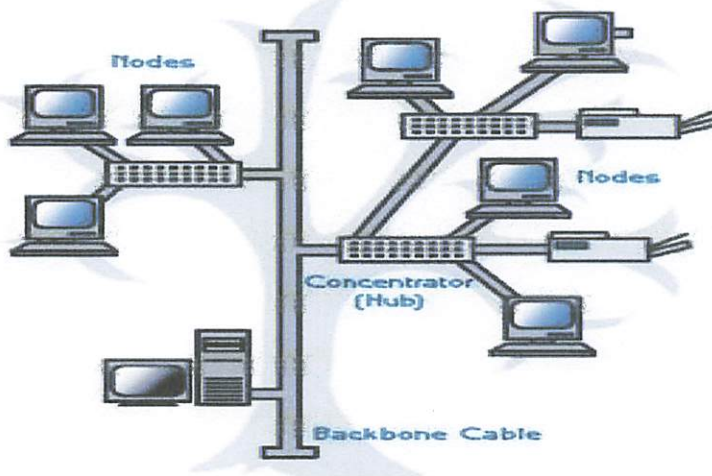
Kekurangan :

1. Menggunakan banyak kabel
2. Ada kemungkinan akan terjadi tabrakan data sehingga dapat menyebabkan jaringan lambat
3. Jaringan sangat tergantung kepada terminal pusat
4. Jaringan memakan biaya tinggi
5. Jika titik komputer pusat terjadi gangguan maka terganggu pula seluruh jaringan.

4. Topologi Tree

Topologi tree ini merupakan hasil pengembangan dari topologi star dan topologi bus yang terdiri dari kumpulan topologi star dan dihubungkan dengan 1

topologi bus. Topologi tree biasanya disebut juga topologi jaringan bertingkat dan digunakan interkoneksi antar sentral. Pada jaringan ini memiliki beberapa tingkatan simpul yang ditetapkan dengan suatu hirarki, gambarannya adalah semakin tinggi kedudukannya maka semakin tinggi pula hirarki-nya. Setiap simpul yang memiliki kedudukan tinggi dapat mengatur simpul yang memiliki kedudukan yang rendah. Data dikirim dari pusat simpul kemudian bergerak menuju simpul rendah dan menuju ke simpul yang lebih tinggi terlebih dahulu



Gambar 2.6 Topologi Tree [2]

Topologi tree ini memiliki kelebihan dan kelemahan yang sama dengan topologi star antara lain:

Kelebihan :

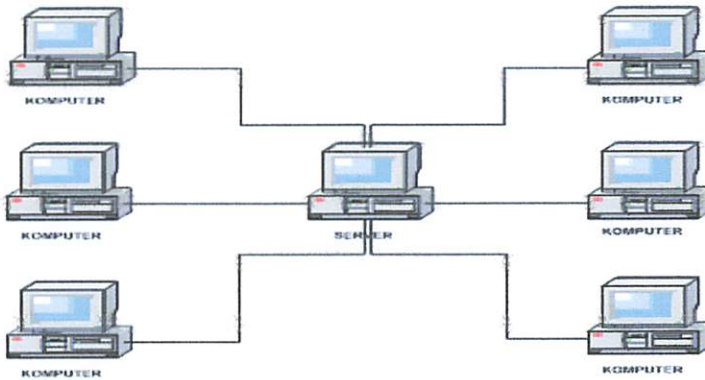
1. Deteksi kesalahan mudah dilakukan
2. Perubahan bentuk suatu kelompok mudah dilakukan dan tidak mengganggu jaringan lain
3. Mudah melakukan control

Kekurangan :

1. Menggunakan banyak kabel
2. Sering terjadi tabrakan data
3. Jirendah akan terganggu juga ka simpul yang lebih tinggi rusak maka simpul yang lebih
4. Cara kerja lambat

5. Topologi Mesh

Topologi Mesh merupakan rangkaian jaringan yang saling terhubung secara mutlak dimana setiap perangkat komputer akan terhubung secara langsung ke setiap titik perangkat lainnya. Setiap titik komputer akan mempunyai titik yang siap untuk berkomunikasi secara langsung dengan titik perangkat komputer lain yang menjadi tujuannya.



Gambar 2.7 Topologi Tree [2]

Kelebihan :

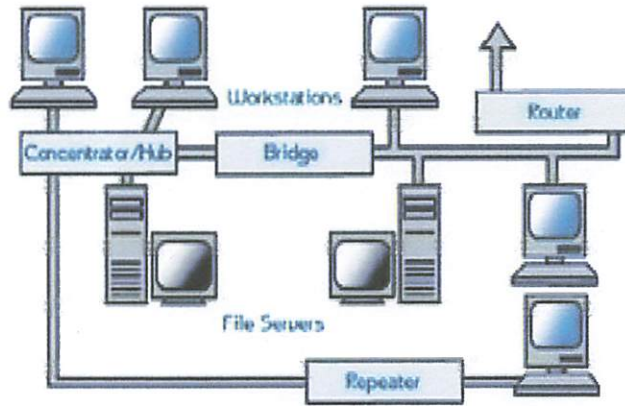
1. Dinamis dalam memperbaiki setiap kerusakan titik jaringan komputer
2. Data langsung dikirimkan ke tujuan tanpa harus melalui komputer lain
3. Data lebih cepat proses pengiriman data
4. Jika terjadi kerusakan pada salah satu komputer tidak akan mengganggu komputer lainnya

Kekurangan :

1. Biaya untuk memasangnya sangat besar.
2. Perlu banyak kabel
3. Perlu banyak port I/O , setiap komputer diperlukan $n-1$ port I/O dan sebanyak $\frac{n(n-1)}{2}$ koneksi. Misalnya ada 4 komputer maka diperlukan kabel koneksi sebanyak $\frac{4(4-1)}{2} = 6$ kabel dan memerlukan $4-1 = 3$ port.
4. Proses instalasi sulit dan rumit

6. Topologi Linear

Topologi ini merupakan perluasan dari dari topologi bus dimana kabel utama harus dihubungkan ke tiap titik komputer menggunakan *T-connector*. Topologi tipe ini merupakan jenis yang sederhana menggunakan kabel RG-58.



Gambar 2.8 Topologi Linear [2]

Kelebihan :

1. Sederhana jaringannya
2. Hemat kabel
3. Mudah untuk dikembangkan

Kekurangan :

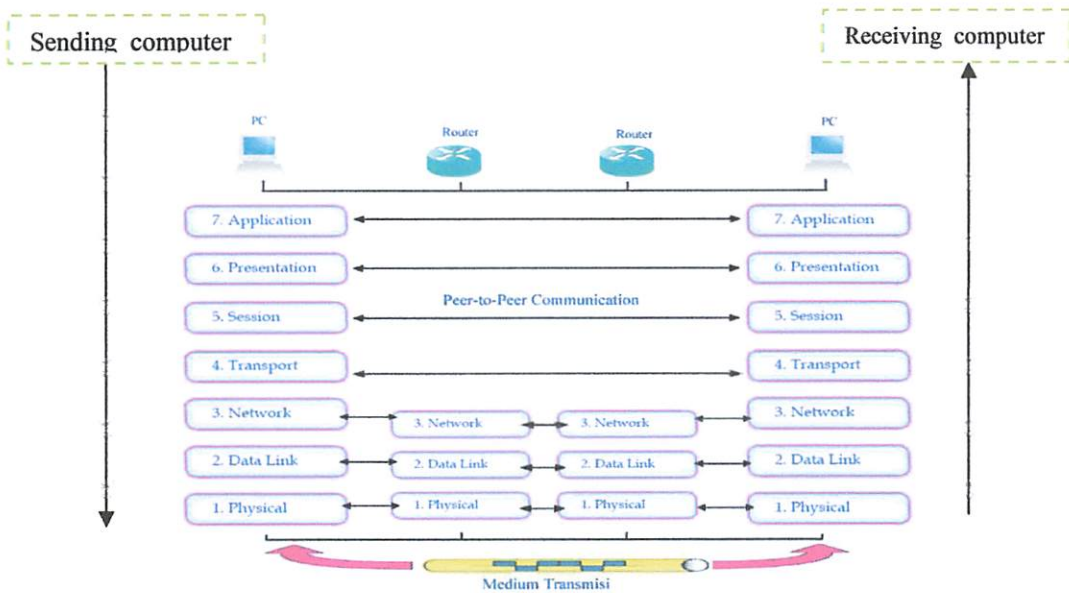
1. Deteksi kesalahan sangat kecil
2. Keamanan kurang terjamin
3. Lalu lintas data tinggi
4. Kecepatan transfer tergantung kepada jumlah pengguna, kecepatan turun jika jumlah pemakai bertambah

2.2 Open Systems Interconnection (OSI)

Pengertian dan Fungsi Layer pada OSI merupakan singkatan dari *Open System Interconnection* adalah standar komunikasi yang diterapkan di dalam jaringan komputer. Standar itulah yang menyebabkan seluruh alat komunikasi dapat saling berkomunikasi melalui jaringan. Dahulu ketika OSI belum digunakan, perangkat komunikasi yang berasal dari vendor berbeda tidak dapat saling berkomunikasi. Alat komunikasi yang diciptakan oleh IBM tidak dapat berkomunikasi dengan vendor lain. Sehingga dibentuklah standard OSI. tujuan

utama dalam penggunaan model OSI adalah untuk membantu designer jaringan memahami fungsi dari tiap layer yang berhubungan dengan aliran komunikasi data. Termasuk jenis-jenis protokol jaringan dan metode transmisi.

OSI terdiri dari tujuh layer yang secara umum terbagi dalam dua kelompok yakni *Upper Layer (Application Layer)* dan *Lower Layer (Transport Layer)*. Layer yang tergolong dalam *Upper Layer* mendefinisikan bagaimana aplikasi pada sebuah host akan berkomunikasi dengan user dan host lainnya. Sedangkan *Lower Layer* mendefinisikan bagaimana data terkirim dari suatu host ke host lainnya.



Gambar 2.9 Lapisan Pada Osi Layer [3]

Model OSI terdiri dari tujuh layer, antara lain :

1. Application

Layer Transport berfungsi untuk memecah data kedalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Lapisan Transport memungkinkan *user node* dan *host node* saling berkomunikasi. Transport Layer bertanggung jawab membagi data menjadi segmen, menjaga koneksi logika “*end-to-end*” antar terminal, dan menyediakan penanganan *error (error handling)*.

2. *Presentation*

Layer *Presentation* adalah layer yang bertanggung jawab dalam memberikan layanan penyajian format data untuk aplikasi-aplikasi pada layer *application* agar tidak terjadi pertukaran data yang tidak diinginkan antara aplikasi satu dengan yang lainnya. Lapisan Penyajian menformat data sehingga dapat disajikan oleh user. *Presentation* Layer bertanggung jawab bagaimana data dikonversi dan diformat untuk transfer data. Contoh konversi *format text ASCII* untuk dokumen, gif dan JPG untuk gambar. Layer ini membentuk kode konversi, translasi data, enkripsi dan konversi.

3. *Session*

Layer *Session* Berfungsi dalam menetapkan, mengelola, dan mengakhiri sesi antara dua host berkomunikasi. Layer ini memberikan layanan ke lapisan *presentation*. Hal ini juga mensinkronkan dialog antara lapisan presentasi dua host dan mengelola pertukaran data mereka. Lapisan *Sessions* bertugas mengatur, mengorganisir dialog dalam pertukaran data. *Session Layer* menentukan bagaimana dua terminal menjaga, memelihara dan mengatur koneksi bagaimana mereka saling berhubungan satu sama lain. Koneksi di layer ini disebut “*session*”.

4. *Transport*

Layer *Transport* berfungsi untuk memecah data kedalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Lapisan *Transport* memungkinkan *user node* dan *host node* saling berkomunikasi. *Transport Layer* bertanggung jawab membagi data menjadi segmen, menjaga koneksi logika “*end-to-end*” antar terminal, dan menyediakan penanganan *error* (*error handling*).

5. *Network*

Layer *Network* berfungsi untuk mendefinisikan alamat-alamat IP, membuat *Header* untuk paket-paket, dan kemudian melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch layer*. Lapisan jaringan menyebabkan lapisan fisik mentransfer frames dari node

kenode. Komponen Network Layer bertanggung jawab menentukan alamat jaringan, menentukan rute yang harus diambil selama perjalanan, dan menjaga antrian trafik di jaringan. Data pada layer ini berbentuk paket.

6. *Datalink Layer*

Datalink berfungsi untuk memberikan pengalamatan fisik dalam hal ini berlawanan dengan pengalaman logis, merubah data menjadi frame dan melakukan *flowcontrol* pada pengiriman data. Lapisan Data Link memformat data menjadi *record* dan mendeteksi kesalahan. Data Link Layer menyediakan link untuk data, memaketkannya menjadi frame yang berhubungan dengan "*hardware*" kemudian diangkut melalui media, *flow control* agar penerima tidak kebanjiran data yang diterima. Komunikasinya dengan kartu jaringan, mengatur komunikasi layer *physical* antara system koneksi dan penanganan *error*. Bridge adalah alat yang digunakan pada suatu jaringan yang berfungsi untuk memisahkan sebuah jaringan yang luas menjadi segment yang lebih kecil. Switch Sebuah alat yang menyaring / filter dan melewatkan (mengijinkan lewat) paket yang ada di sebuah LAN. switcher bekerja pada layer data link (layer 2) dan terkadang di Network Layer (layer 3) berdasarkan referensi OSI Layer.

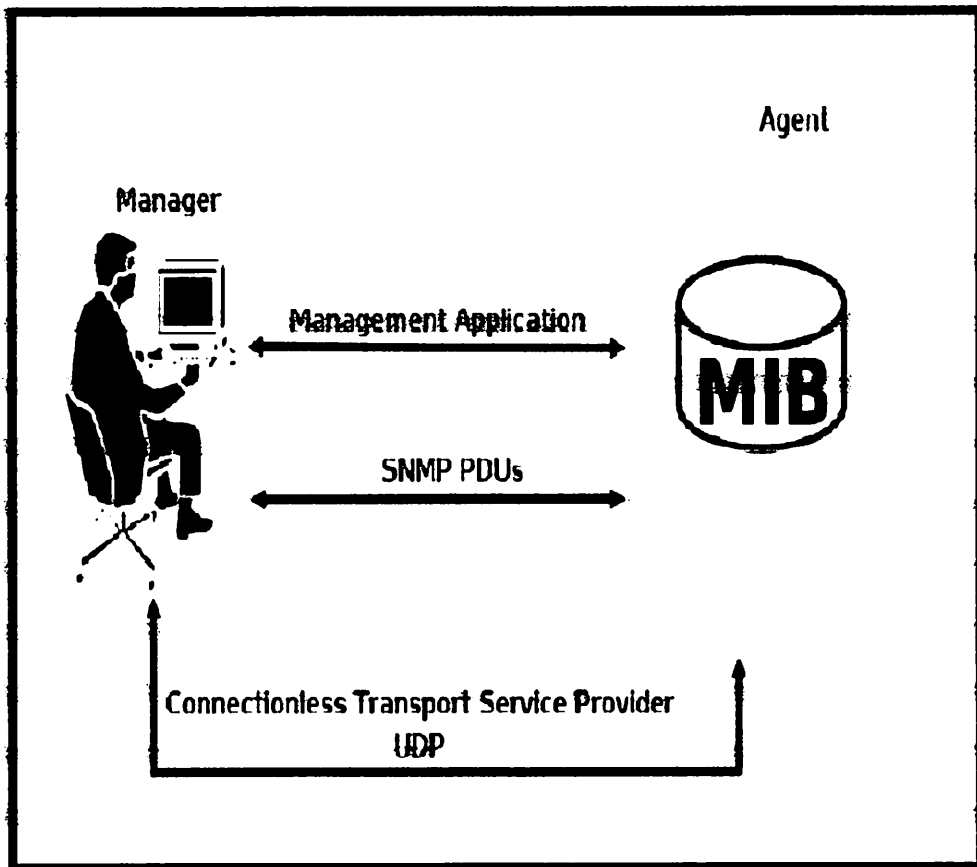
7. *Physhical*

Layer *Physical* berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan topologi dan pengabelan. Lapisan Fisik mentransmisikan data dari satu node ke node lain. Komponen *Physical* Layer bertanggung jawab atas proses data menjadi bit dan mentransfernya melalui media, seperti kabel, dan menjaga koneksi fisik, *transmission rate* menentukan kecepatan pengiriman data antar sistem. HUB adalah sebuah alat yang digunakan dalam jaringan LAN yang bekerja di OSI pada layer *Physical layer*. Hub berfungsi sebagai konsentrator yaitu digunakan untuk menghubungkan 2 atau lebih komputer yang ada pada jaringan LAN meregenerasi/ menghasilkan kembali sinyal yang ditransmisikan pada kabel. Repeater mengijinkan sinyal untuk mengalir diluar batas keterbatasan panjang kabel. Sebuah repeater tidak melakukan

translasi atau filterisasi paket. Contoh protokolnya: 10baseT berfungsi menghubungkan network adapter, 100baseT berfungsi data difastethernet RS232 mengatur mengenai level tegangan, konektor dan aturan komunikasi.

2.3 Simple Network Management Protocol (SNMP)

SNMP merupakan salah satu protokol resmi dari *Internet Protocol suite* yang dibuat oleh *Internet Engineering Task Force (IETF)*. SNMP merupakan contoh dari layer 7 aplikasi yang digunakan oleh *network management system* untuk memonitor perangkat jaringan sehingga dapat memberikan informasi yang dibutuhkan bagi pengelolanya (administrator jaringan).

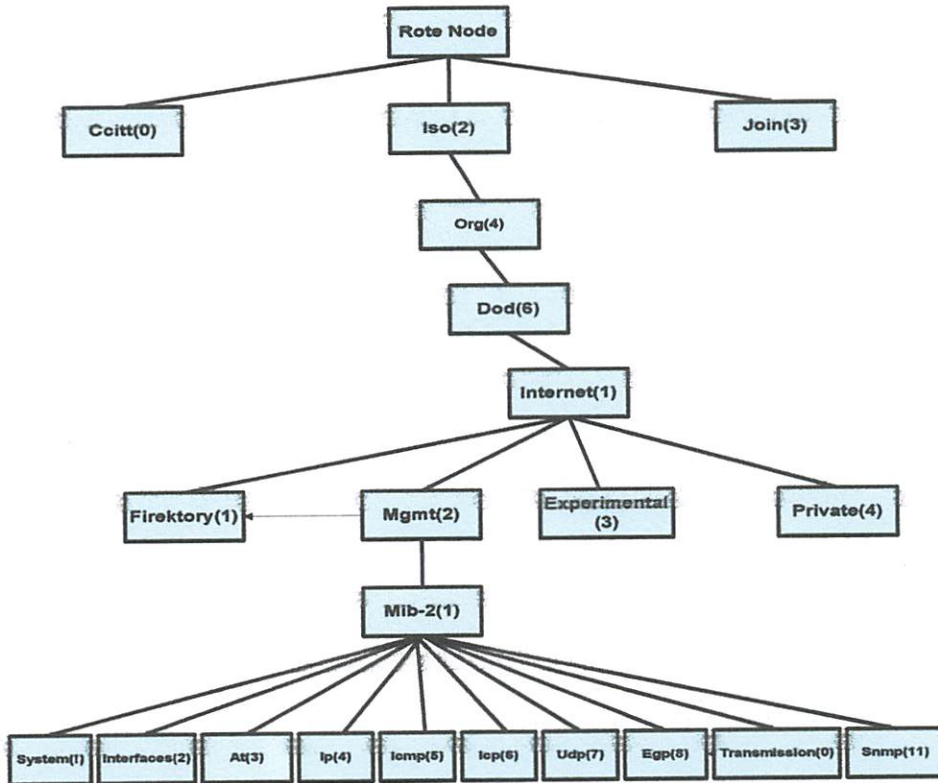


Gambar 2.10 Struktur SNMP [7]

2.3.1. Management Information Base (MIB)

MIB merupakan database yang digunakan untuk manajemen perangkat pada jaringan. Database tersebut berisikan objek entiti dari perangkat jaringan

(seperti router atau switch). MIB dapat dikatakan sebagai struktur basis data variabel dari elemen jaringan yang di kelolanya. Objek pada MIB didefinisikan menggunakan *Abstract Syntax Notation One* (ASN 1), dan diberi nama "*Structure of Management Information Version 2* (SMIV2). Software yang digunakan untuk parsing disebut MIB compiler. RFC yang membahas antara lain RFC1155 – *Structure and identification of Management Information for TCP/IP base internets*, RFC1213 – *Management Information Base for Network Management of TCP/IP-based internets*, dan RFC 1157 – *A Simple Network Management Protocol*. Dalam SNMP, komunikasi yang terjadi antara management station (contoh: console) dengan management object (seperti router, gateway dan switch), menggunakan MIB. *Component* yang berkerja untuk mengambil data disebut SNMP agent, merupakan software yang dapat berkomunikasi dengan SNMP Manager. Mekanisme pendefinisian MIB dalam SNMP menggunakan diagram pohon , dan menempatkan setiap indentifikasi objek (ID) pada suatu lokasi unik pada pohon. Diagram pohon ini dapat di lihat pada gambar 2.2. (Objek Identifier dalam MIB). ID dari objek-objek tersebut mirip dengan nomor telpon yang di organisasikan secara hirarki. Masing – masing objek memiliki angka tertentu yang menunjukkan organisasi tertentu sebagai pemilik. Tingkat puncak digram pohon mendaftar semua organisasi standar yang penting di dunia menurut pandangan ISO, yaitu CCITT (*Consultative Committee For International Telegraph and Telephone*) yang sekarang menjadi ITU (*International Telecommunication Union*), ISO (*International Organization For Standardization*) dan joint-ISO – CCITT. Sebagian besar aktifitas MIB saat ini merupakan bagian dari cabang ISO yang didefinisikan oleh ID 1.3.6.1 yang di kususkan untuk komunikasi internet.



Gambar 2.11 Objek Indentifier Dalam MIB

2.3.2. Arsitektur SNMP

Framework dari SNMP terdiri dari :

❖ *Master Agent*

Master Agent Merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP, fungsinya merespon permintaan dari *SNMP management station*. *Master agent* kemudian meneruskan kepada sub agent untuk memberikan informasi tentang *management* dengan fungsi tertentu. Sebagai contoh, sebuah router dapat menjawab permohonan SNMP dari *management station*. Oleh karenanya sebetulnya berfungsi sebagai server dalam arsitektur client-server atau sebagai daemon dalam terminologi sistem operasi. Sebuah master agen bergantung pada subagen untuk memperoleh oleh informasi manajemen dari sebuah fungsi yang spesifik. Master agen juga sering di sebut sebagai *managed object*.

❖ *Subagent*

Subagent merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP dan mengimplementasikan MIB. Fungsinya

mengumpulkan informasi untuk selanjutnya diproses oleh *management stations*. Subagen adalah sebuah software yang jalan di komponen jaringan yang mampu SNMP yang mengimplementasikan fungsi untuk informasi dan manajemen seperti di definisikan oleh MIB dari subsistem yang spesifik, contoh Ethernet link layer. Beberapa kemampuan subagen adalah: Mengumpulkan informasi untuk managed object. Mengkonfigurasi parameter dari managed object. Merespon kepada permintaan / request dari manager.

❖ *Management Station*

Management Station merupakan *client* dan melakukan permintaan serta mendapatkan *trap* dari *SNMP server*. *Management Station* adalah komponen akhir dari arsitektur SNMP. Fungsinya ekuivalen dengan client di arsitektur client-server. Stasiun manajemen akan mengirimkan request untuk operasi manajemen atas nama administrator jaringan atau aplikasi dan menerima tangkapan dari agen-agen.

2.3.3. Protokol SNMP

Protokol data unit (PDU) dari SNMP (versi 1) antara lain :

Pesan	Uraian
GET REQUEST	Digunakan untuk mendapatkan informasi manajemen
GETNEXT REQUEST	Digunakan secara iteratif untuk mendapatkan sekuen dari informasi manajemen
SET	Digunakan untuk melakukan perubahan terhadap subsistem
TRAP	Digunakan untuk melakukan pelaporan terhadap subsistem Manajemen
GETBULK REQUEST	Iterasi yang lebih cepat untuk mendapatkan informasi
INFORM	Pesan manajer ke manajer yang menjelaskan MIB lokal

Tabel 2.1 Protokol data unit (PDU) dari SNMP

SNMP menggunakan UDP pada port 161 untuk agent dan 162 untuk manager. Manager mengirimkan permintaan terhadap agent pada port 161 dan diterima pada manager pada port 162.

2.3.4. Perkembangan dan penggunaan SNMP

Berikut ini adalah perkembangan dan penggunaan SNMP

1. Version 1

RFC untuk SNMP, dikenal dengan nama *Simple Network Management Protocol* version 1, pada tahun 1988 :

- RFC 1065 – Structure and identification of management information for TCP/Ipbased internets
- RFC 1066 – Management information base for network management of TCP/Ipbased internets
- RFC 1067 – A Simple Network Management Protocol

Kemudian menjadi kadaluwarsa dengan digantikan dengan :

- RFC 1155 – Structure and identification of management information for TCP/Ipbased internets
- RFC 1156 – Management information base for network management of TCP/Ipbased internets
- RFC 1167 – A Simple Network Management Protocol

Versi 1 memiliki kelemahan pada sistem autentifikasi karena mengirimkan password secara plain text.

2. Version 2

Versi 2 ini banyak yang tidak menggunakan dikarenakan ketidakcocokan framework. Simple Network Management Protocol version 2 (RFC 1441 – RFC 1452) dan juga dikenal sebagai SNMP v2. Diperkenalkan GETBULK sebagai alternatif dari GETNEXT. Dikenalkan juga *Community-Based Simple Network Management Protocol* version 2 atau yang disebut SNMP v2c sebagai pengganti sistem autentikasi *User-Based Simple Network Management Protocol* version 2, atau SNMP v2u yang digunakan untuk memperbaiki keamanan dari SNMP v1.

3. Version 3

Versi ini didefinisikan pada RFC 3411 – RFC 3418 yaitu Simple Network Management Protocol version 3, dikeluarkan pada tahun 2004. Pada prakteknya SNMP bisa menggunakan versi SNMPv1, SNMPv2c, atau SNMPv3. Dijabarkan pada RFC 3584 – Coexistence between Version 1, Version 2, and Version 3 of the *Internet-Standard Network Management Framework*.

2.3.5. Mengimplementasikan SNMP

Banyak informasi yang bisa ditampilkan, misal performance, load dan error rate dari suatu jaringan seperti router atau switch. Kemudian dengan fungsi khusus, informasi yang didapat diolah dalam bentuk grafik. Contoh aplikasi Multi Router Traffic Grapher dan Cacti Analog dengan perkembangan dan pengimplementasian SNMP yang berubah dari versi satu ke versi lainnya.

2.4 Perkembangan Android

Android adalah salah satu sistem operasi untuk telepon seluler yang berbasis Linux. Pada awalnya dikembangkan oleh Android Inc. (sebuah perusahaan telepon seluler yang berada di California, AS), pada bulan Juli 2000. Rumor yang berkembang adalah bahwa Google akan memasuki pasar telepon seluler, dikarenakan kerjasama yang dilakukan antara Google dengan Android Inc.

Sebuah tim yang memiliki tugas untuk mengembangkan program perangkat seluler yang didukung oleh kernel Linux akhirnya dibentuk, hal ini menjadikan rumor yang beredar semakin kuat bahwa memang Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Untuk mempermudah usaha Google untuk mengembangkan produk Android, maka Google membeli Android Inc dan serta dibentuklah “Open Handset Alliance” yang tergabung atas 34 perusahaan software, hardware dan telekomunikasi seperti : Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia dan lainnya. Sekitar September 2007, Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar yang menggunakan Android pada sistem operasinya. Telepon 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 ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer 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. Telepon seluler pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008.



Gambar 2.12 Gambar Logo Android [6].

Berikut adalah sejarah dan fitur sistem operasi Android semua versi:

1. Android Versi 1.0 (Apple pie)

Android 1.0 dirilis pada tanggal 23 September 2008 yang memiliki kode nama Apple pie

serta ukuran layar 320×480 HVGA.

2. Android Versi 1.1 (Banana bread)

Android 1.1 dirilis pada tanggal 9 Februari 2009 yang memiliki kode nama Banana bread serta ukuran layar 320×480 HVA.

3. Android 1.5 (Cupcake)

Android 1.5 dirilis pada tanggal 30 April 2009 yang memiliki kode nama Cupcake. Terdapat beberapa pembaruan termasuk juga penambahan beberapa

fitur yakni kemampuan merekam dan menonton video, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon seluler, dukungan Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

4. Android 1.6 (Donut)

Android 1.6 dirilis pada tanggal 15 September 2009 yang memiliki kode nama Donut. Versi ini perbaikan dari Android Cupcake dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel); pengadaan resolusi VWGA.

5. Android versi 2.0/2.1 (Eclair)

Android versi 2.0/2.1 dirilis pada tanggal 3 Desember 2009 yang memiliki kode nama Eclair. perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

6. Android versi 2.2 (Froyo)

Android versi 2.2 dirilis pada tanggal 20 Mei 2010 yang memiliki kode nama Froyo. Froyo akan memperkenalkan peningkatan kecepatan dengan JIT optimalisasi dan Chrome V8 mesin JavaScript, dan menambahkan Wi-Fi hotspot dan dukungan Adobe Flash.

7. Android versi 2.3 (Gingerbread)

Android versi 2.3 dirilis pada tanggal 6 Desember 2010 yang memiliki kode nama Gingerbread. Android Gingerbread adalah android yang mempermudah pengguna antarmuka, meningkatkan keyboard lunak dan fitur copy / paste, dan menambahkan dukungan untuk *Near Field Communication*. pada versi ini Android telah melakukan update sebanyak 5 kali yang dirilis pada tanggal yang berbeda-beda serta penambahan difitur-fitur baru, mereka masing adalah Android 2.3.3, Android 2.3.4, Android, 2.3.5, Android 2.3.6, dan terakhir Android 2.3.7

8. Android versi 3.0 (Honeycomb)

Android versi 3.0 dirilis pada tanggal 15 Juli 2011 yang memiliki kode nama Honeycomb. Di versi ini Android akhirnya jalan pada sebuah Tablet, dan tablet pertama yang menggunakan sistem operasi Honeycomb adalah Motorola Xoom.

9. Android versi 4.0 (Ice Cream Sandwich)

Android versi 4.0 dirilis pada tanggal 19 Oktober 2011 yang memiliki kode nama Ice Cream Sandwich. Di versi Android menyempurnakan seluruh GUI yang dibangun dan penambahan fitur seperti *Facial recognition (Face Unlock)*, *UI use Hardware acceleration*, *Better voice recognition (dictating/Voice typing)*, *Web browser, allows up to 16 tabs*, *Updated launcher (customizable)*, *Android Beam app to exchange data through NFC*, *Resizable widgets*.

10. Android 4.1 Jelly Bean (API level 16)

Android versi 4.1 Jelly Bean dirilis pada 9 Juli 2012 yang memiliki Antarmuka pengguna yang lebih halus, Peningkatan aksesibilitas, Papan ketik yang bisa dimodifikasi oleh pengguna, *Shortcut dan widget* secara otomatis bisa disusun ulang atau diatur ukurannya.

11. Android 4.4 KitKat (API level 19)

Android 4.4 KitKat dirilis pada 31 Oktober 2013 yang memiliki pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan, optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah, *NFC Host Card Emulation* sebagai emulator kartu pintar, kerangka kerja akses penyimpanan untuk mengambil konten dan dokumen dari sumber lain, perekam aktivitas layar yang terintegrasi, dukungan Bluetooth *Message Access Profile (MAP)*.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug.

Android memiliki berbagai keunggulan sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (*open source*)

sehingga pengguna bisa membuat aplikasi baru di dalamnya. Android memiliki aplikasi native Google yang terintegrasi seperti pushmail Gmail, Google Maps, dan Google Calendar.

Para penggemar open source kemudian membangun komunitas yang membangun dan berbagi Android berbasis firmware dengan sejumlah penyesuaian dan fitur-fitur tambahan, seperti FLAC lossless audio dan kemampuan untuk menyimpan download aplikasi pada microSD card. Mereka sering memperbaharui paket-paket firmware dan menggabungkan elemen-elemen fungsi Android yang belum resmi diluncurkan dalam suatu carrier-sanction firmware.

Berikut sejumlah vendor yang mengeluarkan telepon seluler berbasis Android

1. HTC
2. Samsung
3. LG
4. Motorola
5. Sony
6. Acer
7. Dell

2.5 Software Development Kit (SDK) Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent).



Gambar 2.13 Logo *Software Development Kit Eclipse* [6]

... Google ...

... 1997 ...

- 1. ...
- 2. ...
- 3. ...
- 4. ...
- 5. ...
- 6. ...
- 7. ...
- 8. ...
- 9. ...
- 10. ...

... (continued) ...



Figure 1: ...

Berikut ini adalah sifat dari Eclipse:

1. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. **Mult-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source 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 membuat komponen yang disebut *plug-in*.

2.5.1. Sejarah

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.

2.5.2. Arsitektur

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah dipasang (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 JDIT (*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 adalah:

- a) terbuka (open),
- b) mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik.

Eclipse tidak saja untuk mengembangkan program Java, tetapi juga untuk berbagai macam keperluan. Perluasan apapun cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat plug-in CDT (*C/C++ Development Tools*) yang dapat dipasang di Eclipse untuk Eclipse menjadi perangkat untuk pengembangan C/C++.

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. Salah satu situs yang menawarkan *plug-in* yang gratis adalah *Eclipse downloads by project*.

2.5.3. Histori (versi-versi) Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

BAB III

PERANCANGAN DAN ANALISA SISTEM

3.1 Deskripsi Sitem

Deskripsi sistem adalah gambaran tentang sistem yang akan dibangun untuk *Network monitoring* / memonitoring jaringan menggunakan aplikasi di smart phone android. Aplikasi ini dibangun sebagai system untuk membantu administrator dalam memonitoring jaringan yang dikelolanya. *Network monitoring* dapat menggambarkan bagaimana suatu sistem secara konstan memonitor kondisi jaringan seperti *router* maupun *device* lainnya. Apabila pada suatu sistem jaringan terjadi suatu error / down / tidak stabil maka secara otomatis keadaan tersebut akan diketahui *network administrator* atau *system administrator*. Maka dalam hal ini, seorang *network administrator* memiliki andil dalam memonitor atau memantau jaringan agar jaringan selalu berada dalam kondisi optimal. Monitoring jaringan komputer dapat berupa suatu koleksi informasi (data) yang merupakan salah satu fungsi dari manajemen jaringan yang berguna untuk menganalisa apakah suatu jaringan masih cukup layak untuk digunakan untuk berkomunikasi atau memerlukan tambahan kapasitas. Hasil dari monitoring dapat membantu jika admin ingin mendesain ulang jaringan yang telah ada. Banyak hal dalam jaringan yang bisa di monitoring, salah satu diantaranya yaitu *load traffic* jaringan yang lewat pada sebuah router atau interface komputer.

Proses awal yang dilakukan adalah menentukan ip jaringan yang akan di minitoring. Monitoring jaringan adalah salah satu fungsi dari management yang berguna untuk menganalisa apakah jaringan masih cukup layak untuk digunakan atau perlutambahan kapasitas .Dengan aplikasi ini sangat membatu admin jaringan untuk mematau dan memonitoring aktifitas jaringan secara berkala. Proses dalam analisis data dapat berupa pemilihan data dari sejumlah data yang telah terkumpul atau bias juga berupa manipulasi data sehingga diperoleh informasi yang diharapkan. Sumber data dapat berupa network traffic, informasi mengenai hardware, dan lain sebagainya. Sedangkan tahap menampilkan data hasil monitoring menjadi informasi yang berguna di dalam pengambilan

keputusan atau kebijakan terhadap sistem yang sedang berjalan dapat berupa sebuah tabel, gambar, kurva, atau animasi. Aksi yang terjadi diantara proses-proses yang ada di dalam sebuah sistem monitoring adalah berbentuk service, yaitu suatu proses yang terus-menerus berjalan pada interval waktu tertentu. Proses yang dijalankan dapat berupa pengumpulan data dari objek yang dimonitor atau melakukan analisis data yang telah diperoleh dan menampilkannya. Sebuah sistem monitoring melakukan proses pengumpulan data mengenai dirinya sendiri dan melakukan analisis terhadap data-data tersebut dengan tujuan untuk memaksimalkan seluruh sumber daya yang dimiliki. Untuk mendesain sebuah sistem monitoring jaringan dibutuhkan suatu mekanisme yang dapat memberikan informasi-informasi yang dapat menggambarkan keadaan suatu jaringan terhubung atau tidak. Mekanisme untuk mendapatkan informasi – informasi tersebut dapat diperoleh dengan pemanfaatan SNMP sebagai protokol yang mengirim permintaan dari stasiun manajemen keagen pada device computer, sedangkan IP adalah protokol yang membawa informasi tersebut dari mana dan harus kemana informasi harus disampaikan. Aplikasi ini didesain untuk memonitoring aktifitas dari suatu jaringan komputer.

3.2 Analisa Sistem

Tahapan analisis sistem ini akan membahas mengenai permasalahan yang dihadapi, analisa masalah, analisa data, analisa terhadap perangkat lunak, perangkat keras serta user yang akan menggunakan aplikasi ini, sehingga dapat mempermudah dalam melakukan perancangan dan implementasi aplikasi ini.

3.2.1 Analisa Masalah

Masalah yang dihadapi dalam membangun aplikasi ini adalah bagaimana aplikasi yang dibangun dapat memberikan kemudahan kepada seseorang dalam memonitoring jaringan.

3.2.3 Analisa Perangkat Lunak

Aplikasi yang akan dirancang adalah suatu aplikasi yang dapat membantu admin jaringan untuk memonitoring jaringan yang dikelolanya, dimana perangkat lunak yang dibutuhkan akan dianalisa sebagai berikut:

1. Android sebagai Operasi Sistem yang digunakan untuk menjalankan Aplikasi.
2. Software Development Kit (SDK) Eclipse sebagai aplikasi yang digunakan dalam membangun aplikasi.

3.2.4 Analisa Perangkat Keras

Perangkat keras atau hardware merupakan salah satu hal yang harus kita perhatikan, karena tanpa hardware yang memenuhi isyarat, aplikasi yang akan kita bangun tidak akan bias berjalan. Berikut spesifikasi perangkat keras yang digunakan untuk membangun aplikasi ini:

1. Memory 512 MB
2. Processor Intel Pentium IV 2.4MHz
3. Minimum Hard disk 5 GB
4. Video Card 512 MB
5. HandPhone dan Kabel Data

Sedangkan spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan untuk dapat menjalankan aplikasi yang akan dibangun dengan baik adalah sebagai berikut:

1. Minimal Empat Pasang Komputer
2. Jaringan Internet
3. Hand Phone (Android)

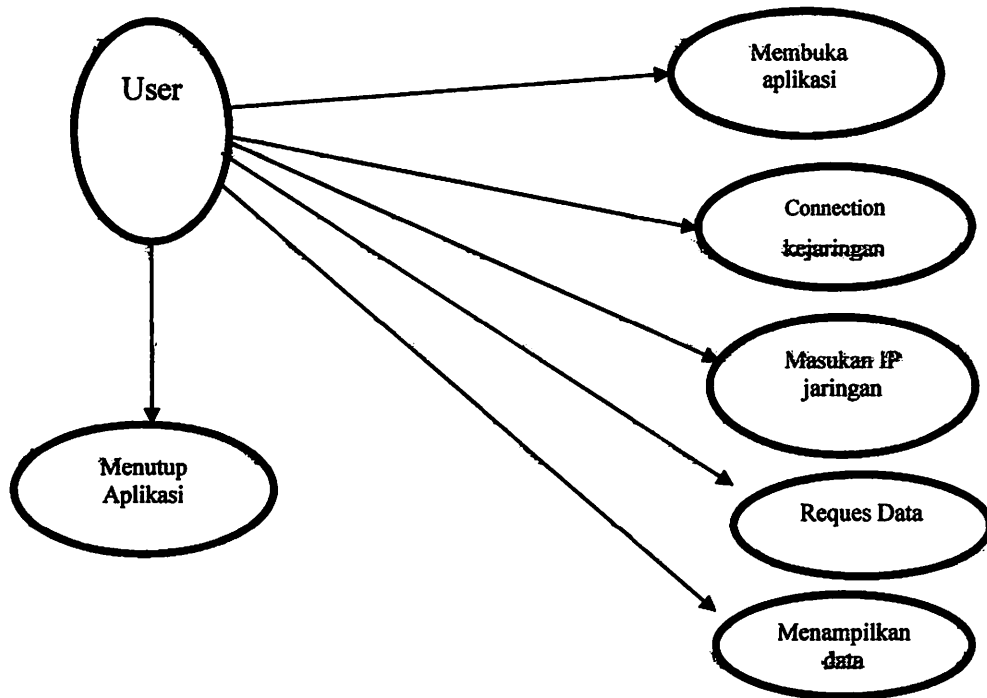
3.1.5 Analisa User

Analisis user yang dapat menggunakan aplikasi monitoring jaringan yang sudah terbiasa menggunakan aplikasi telepon seluler. Analisis user digunakan untuk mempermudah user berinteraksi dengan aplikasi yang akan dibuat, adapun user yang dapat menggunakan aplikasi monitoring jaringan yang sudah terbiasa menggunakan aplikasi telepon seluler.

3.3. Tahap Perancangan

Pada tahap ini dilakukan perancangan aplikasi yang terdiri dari rancangan use case diagram, rancangan diagram alir, desain atau interface aplikasi mobile.

3.3.1 Rancangan Use Case Diagram

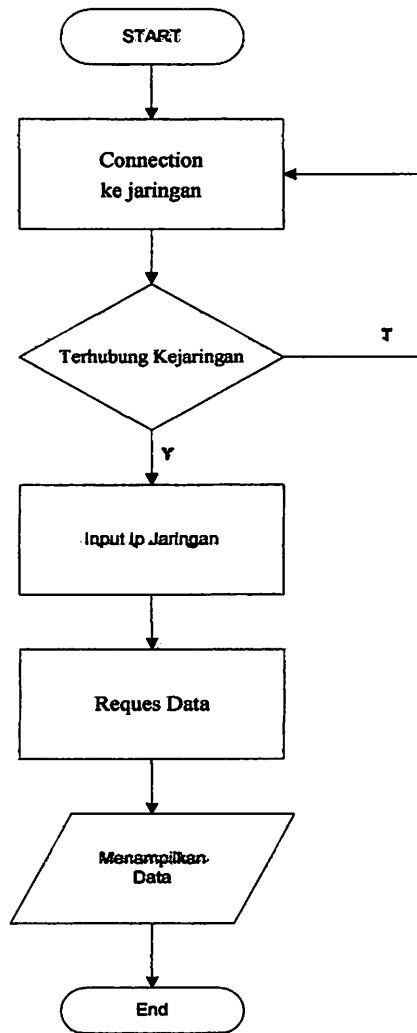


Gambar 3.1 Use Case Diagram

Use case diagram mendefenisikan fitur – fitur yang terdapat dalam system. Gambar 3.1 menunjukkan adanya interaksi antara actor dengan system. Actor yang berperan adalah user yang menjalan kanaplikasi, sedangkan yang melakukan monitoring adalah smartphone android.

3.3.2 Rancangan Diagram Alir

Diagram Alir (Flowchart) adalah gambaran secara grafik yang terdiri dari simbol-simbol dari algoritma-algoritma dalam suatu program, yang menyatakan arah dari alur program. Diagram Alir (Flowchart) digunakan untuk membantu menganalisa untuk memecahkan masalah dalam program yang di buat. Berikut adalah rancangan diagram alir yang di buat untuk menjelaskan cara kerja aplikasi

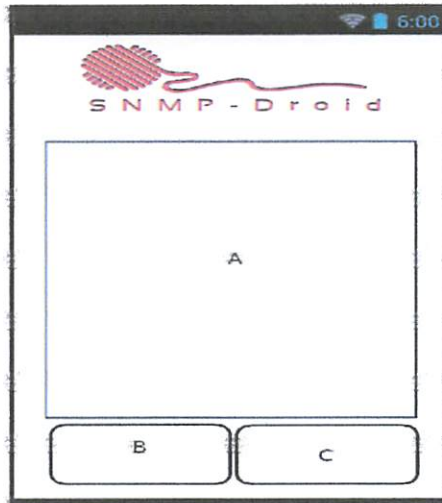


Gambar 3.2Rancangan Diagram Alir

Dari gambar 3.2 Diagram alir system monitoring, dapat dilihat dengan jelas cara kerja aplikasi yaitu user sebagai pengaktif aplikasi di smartphone selanjutnya system menghubungkan ke jaringan dengan menggunakan jaringan wifi sebagai perantara ke jaringan yang akan dimonitoring. Setelah terhubung, masukan alamat Ip jaringan yang akan dimonitoring, selanjutnya system secara otomatis mengrequest data data objek yang dimonitoring untuk mendapatkan informasi jaringan. Setelah data di dapat, dianalisa kemudian di tampilkan.

3.3.3 Desain Atau Interface Tampilan Awal

Desain atau nnterface tampilan awal dibuat untuk memberikan gambaran tampilan ketika aplikasi di buka. Adapun desain atau interface tampilan awal di lihat pada gambar 3.3



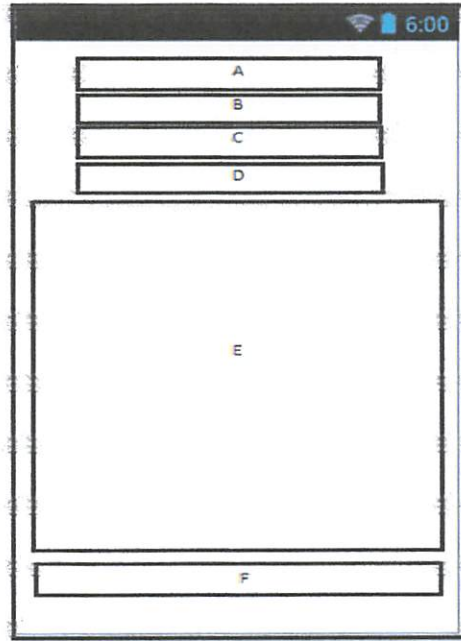
Gambar 3.3Desainatau Interface TampilanAwal

Keterangan :

- A. Tampilan Network Device Dan Ip Address
Menampilkan nama network ip address jaringan yang terhubung dengan aplikasi.
- B. Exit App
Fungsi Tombol Exit App adalah untuk keluar dari aplikasi.
- C. Start Tes
Fungsi Tombol Start Tes adalah untuk menjalankan aplikasi monitoring setelah aplikasi terhubung dengan jaringan.

3.3.4 Desain Atau Interface Tampilan Hasil

Desain atau interface tampilan hasil tujuan dibuat untuk memberikan gambaran tampilan hasil monitoring. Adapun rancangannya dapat di lihat pada gambar 3.4.



Gambar 3.4 Desain Atau Interface Tampilan Hasil

Keterangan :

A. Button “Monitoring Devices”

Fungsinya untuk mengetahui daftar perangkat dan daftar alamat ip jaringan

B. Button “Trafik Jaringan”

Fungsinya untuk mengetahui trafik jaringan

C. Button “Bandwidth Jaringan”

Fungsinya untuk mengetahui pengguna bandwidth jaringan

D. Button “Ip Address List”

Fungsinya untuk mengetahui ip address pengguna yang terhubung ke jaringan

E. Tampilan Hasil Monitoring

F. Button “Clear”

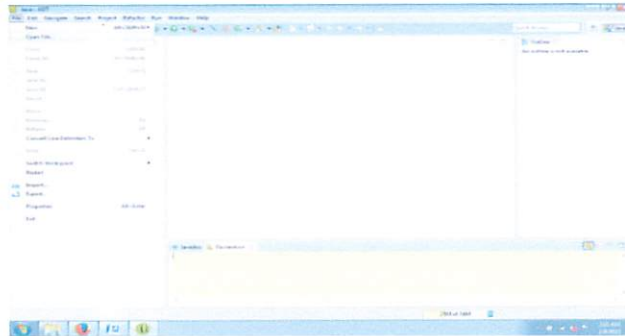
Fungsi Tombol Clear adalah untuk menghapus data hasil monitoring

3.3.5 Perancangan Android *Project* Pada *Eclipse*

Untuk membuat project aplikasi android , semua step-step yang dibutuhkan seperti, instalasi SDK, emulator, dan juga plugin android di eclipse, sudah selesai

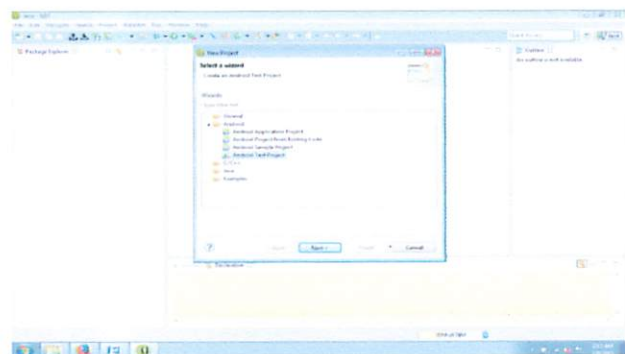
terintal. Setelah semua itu dipersiapkan, berikut ini langka-langka membuat project baru, caranya sebagai berikut:

1. Jalankan eclipse, lalu menuju ke File » New » Project



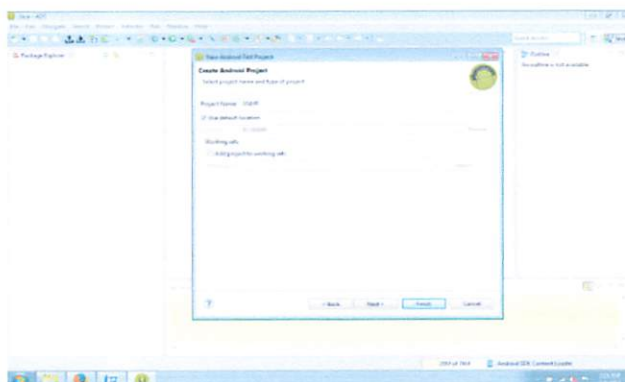
Gambar 3.5 Tampilan Eclipse

2. Pilih Android Project, lalu tekan Next >



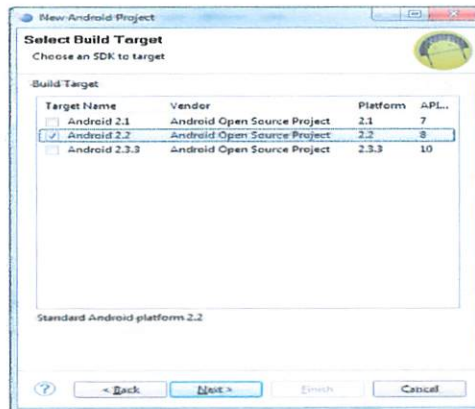
Gambar 3.6 Tampilan Android Project

3. Isikan nama project yang anda inginkan dan di mana anda ingin menyimpan source project tersebut, lalu tekan Next >



Gambar 3.7 Tampilan Nama Project

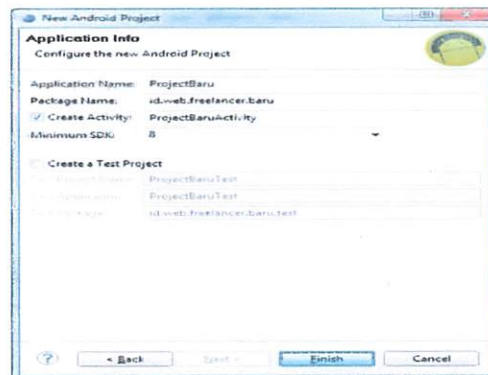
4. Pilih build target yang anda inginkan.



Gambar 3.8 Tampilan Select Buid Target

5. Isikan beberapa info yang dibutuhkan diantaranya:

- Application Name: Nama project yang anda inginkan
- Package Name: Akan digunakan oleh android market, sebagai unique identifier dari aplikasi anda
- Create Activity: Langsung buat satu activity utama (yang akan langsung dijalankan saat aplikasi dibuka di device)
- Minimum SDK: target platform paling kecil yang akan disupport aplikasi anda

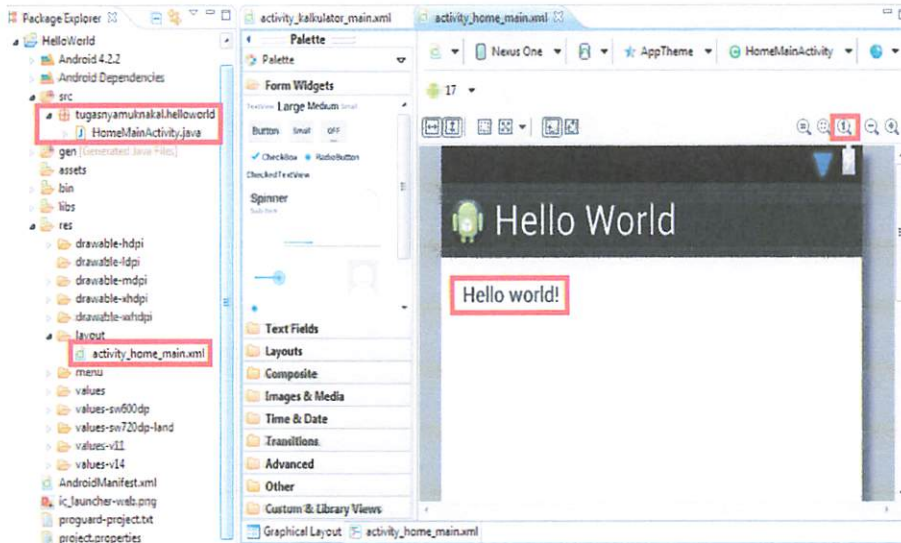


Gambar 3.9 Tampilan Application Info

Setelah semua itu diisi, selanjutnya tekan Finish

6. Ket:

HomeMainActivity.java sebagai tempat penulisan code program nantinya
activity_home_main.xml sebagai pengaturan layout
Hello world adalah TextView standard stiap kali membuat program
1 digunakan untuk zoom pada layout untuk mempermudah design.



Gambar 3.10 Home Main Activity.java

7. Pembuatan Splash Screen

Splash screen merupakan tampilan awal aplikasi sebelum masuk pada layar menu, dengan kode program sebagai berikut:

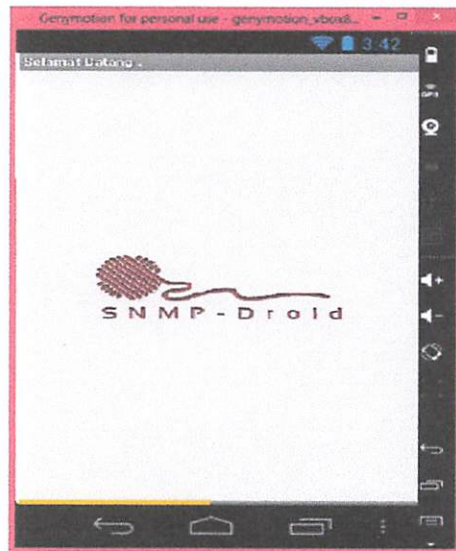
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/relativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#fff0f0" >

    <ImageView
        android:id="@+id/foto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:src="@drawable/Logo" />

    <ProgressBar
        android:id="@+id/progressBar1"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="5dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Pada script "Vertical" menandakan layout yang akan kita buat dengan posisi vertical. Lalu pada "@drawable/logo" kita akan menampilkan gambar yang akan muncul pada *Interface Please Wait*. Untuk menampilkan gambar tersebut simpan gambar tersebut pada folder drawable yang terdapat pada folder res. Untuk menampilkan gambar tersebut simpan gambar tersebut pada folder drawable yang terdapat pada folder res. Sedangkan android:layout_centerVertical="true" berfungsi untuk mengatur letak gambar agar berada pada posisi tengah pada layar.



Gambar 3.11 Tampilan *Splash Screen*

Setelah selesai Pembuatan *Layout Splash Screen*, langkah selanjutnya adalah pembuatan kode program *Splash Screen*

```
public class SplashScreenActivity extends Activity {
    ProgressBar pbMain;
    //tampilan utama
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        setTitle("Selamat Datang..");

        final int welcomeScreenDisplay =100;

        //loading
        pbMain = (ProgressBar) findViewById(R.id.progressBar1 );
        pbMain.setMax(welcomeScreenDisplay);

        Thread threads = new Thread() {

            int wait = 0;
```

```

@Override
public void run() {
    try {
        super.run();
        //proses loading
        while (wait < welcomeScreenDisplay) {
            sleep(20);
            wait += 1;
            //update loading
            pbMain.setProgress(wait);
        }
    } catch (Exception e) {
        System.out.println("Error:" + e);
    } finally {
        newIntens();
    }
}
};
threads.start();
}
//tampil pesan
public void t(String text) {
    Toast.makeText(getApplicationContext(), text,
Toast.LENGTH_LONG).show();
}
//menjalankan tab utama
protected void newIntens() {
    // TODO Auto-generated method stub
    startActivity(new Intent( this,ConnActivity.class ));
    finish();
}
}

```

Pada penggalan script terakhir **public void** di sini kita membuat sebuah inten baru, dimana setelah splash screen ini. Otomatis akan membuka *class java* baru, sedangkan splash screen sendiri otomatis akan berakhir.

8. Pembuatan Menu Tampilan Awal

Selanjutnya pembuatan layar menu Tampilan Awal, dengan kode program sebagai berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <Button
        android:id="@+id/button1"
        android:layout_width="0dp"

```

```

android:layout_height="wrap_content"
android:layout_weight="1"
android:drawableTop="@android:drawable/ic_menu_info_details"
android:text="LAN Connection" />

```

```
<Button
```

```

    android:id="@+id/button2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:drawableTop="@android:drawable/ic_menu_more"
    android:text="Result" />

```

```
</LinearLayout>
```

Pada script `android:text="LAN Connection" />` bertujuan untuk menghubungkan aplikasi dengan jaringan menggunakan jaringan *Lan*. Setelah selesai maka akan muncul tampilan seperti ini:



Gambar 3.12 Tampilan Awal

Pada `menu.java` ini kita mencoba untuk menjalankan interface yang telah kita buat pada `main.xml`, dengan mengetikkan script berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="match_parent"

```

```
android:layout_height="wrap_content"  
android:src="@drawable/Logo" />
```

```
<ScrollView  
    android:id="@+id/scrollView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:padding="10dp" >
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >
```

```
<TextView  
    android:id="@+id/TextView01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="SSID :"
```

```
android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/edA"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:enabled="false" />
```

```
<TextView  
    android:id="@+id/TextView02"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="IP Adres :"
```

```
android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/edC"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:enabled="false" />
```

```
<TextView  
    android:id="@+id/TextView04"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Gateway:"
```

```
android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/edD"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"
```

```

        android:enabled="false" />

<TextView
    android:id="@+id/TextView03"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/target_host"

android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/edHost"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:text="192.168.56.10" />

<TextView
    android:id="@+id/TextView05"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/target_port"

android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/edPort"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:text="@string/port_hint" />

<TextView
    android:id="@+id/TextView06"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/snmp_version"

android:textAppearance="?android:attr/textAppearanceMedium" />
<Spinner
    android:id="@+id/spinnerVersion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/version_array" />

<TextView
    android:id="@+id/TextView07"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/community_string"

android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/edCom"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```

        android:ems="10"
        android:text="public" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/btnA"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="100dp"
        android:layout_height="78dp"
        android:layout_weight="0.99"

        android:drawableTop="@android:drawable/ic_menu_close_clear_cancel"
        android:text="Exit App" />

    <Button
        android:id="@+id/btnB"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="100dp"
        android:layout_height="match_parent"
        android:layout_weight="1.00"

        android:drawableTop="@android:drawable/ic_menu_rotate"
        android:text="Start Test" />
    </LinearLayout>
</LinearLayout>

</ScrollView>

</LinearLayout>

```

Disini penulis membuat dua buah pilihan yakni exsist app dan start test. Jika kita memilih start test maka kita akan masuk ke kelas java baru dengan nama start test.class. Sedangkan jika kita memilih exsist app maka kita akan keluar dari aplikasi. Jika kita memilih keluar maka kita akan keluar dari aplikasi ini.

9. Pembuatan Menu Tampilan hasil

Tampilan hasil tujuan dibuat untuk metampilan hasil monitoring setelah pengguna mengklik button Start Test dengan kode program sebagai berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="10dp"
    android:gravity="center"

```

```

android:layout_gravity="center"
>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnMonitor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Monitor Devices" />

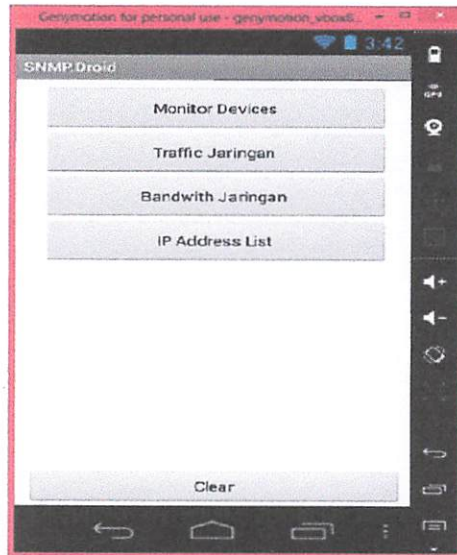
    <Button
        android:id="@+id/btnTraiff"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Traffic Jaringan" />

    <Button
        android:id="@+id/btnBand"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Bandwith Jaringan" />

    <Button
        android:id="@+id/btnFlow"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="IP Address List" />
</LinearLayout>
</LinearLayout>

```

Disini penulis membuat 4 buah pilihan yakni Monitor Devices, Traffic Jaringan, Bandwith Jaringan, dan IP Address List. Jika kita memilih button Monitor Devices maka kita akan masuk ke kelas java baru dengan nama Monitor Devices.class. Jika kita memilih button Traffic Jaringan maka kita akan masuk ke kelas java baru dengan nama Traffic Jaringan.class dan begitu juga dengan button yang lain. Setelah mengetikkan kode program tersebut maka akan muncul layar sebagai berikut:



Gambar 3.13 Tampilan Hasil

Pada menu.java ini kita mencoba untuk menjalankan interface yang telah kita buat pada main.xml, dengan menyetikkan script berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <include layout="@Layout/hasilmenu"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.62"
    android:orientation="vertical" >

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical" >

            <TextView
                android:id="@+id/txtA"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Small Text"

                android:textAppearance="?android:attr/textAppearanceSmall"
```

```

        android:textColor="#FFFF0000"
        android:textSize="15dp" />
    </LinearLayout>
</ScrollView>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnA"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Clear" />
</LinearLayout>

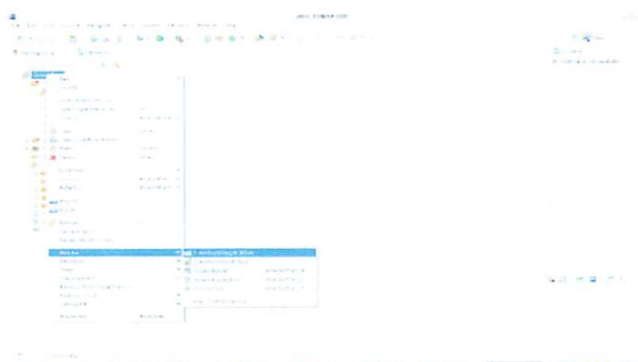
```

Pada RadioGroup terdapat script `android:orientation="vertical"` yang berfungsi untuk mengatur letak RadioButton secara vertical. Maka hasil menu akan berderet dari atas ke bawah. Pada script `android:text="Clear" />` berfungsi untuk menghapus hasil monitoring.

3.6 Uji Coba Aplikasi

Setelah proses pembuatan aplikasi selesai, langkah terakhir yang harus dilakukan adalah uji coba aplikasi pada komputer. Sebelum melakukan uji coba, penulis memastikan terlebih dahulu apakah program tersebut terdapat kesalahan atau tidak. Jika semua coding sudah benar, maka siap untuk dilakukan uji coba.

Pilih package `SNMP.Droid` yang ada pada workspace, kemudian klik kanan, pilih `run as, Android Application`.



Gambar 3.14 Tampilan menu Run As Application

Setelah proses kompilasi program berhasil maka akan muncul emulator untuk menampilkan aplikasi yang telah dibuat.



Gambar 3.15 Tampilan Menu Emulator

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.2 Implementasi Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android

Pembuatan aplikasi mobile remote network monitoring menggunakan mobile device berbasis android di lakukan dengan menerapkan hasil desain yang telah di buat ke dalam bahasa pemrograman java yakni Action script sehingga prosedur-prosedur yang di buat dapat menghasilkan keluaran seperti yang di harapkan.

4.2 Pengujian Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android

Dalam pembuatan aplikasi mobile remote network monitoring menggunakan mobile device berbasis android dilakukan juga tahapan pengujian aplikasi untuk mengetahui fitur yang di gunakan untuk menjalankan aplikasi mobile remote network monitoring menggunakan mobile device berbasis android

4.3 Tampilan Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android

Dalam pengujian Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android dapat dilihat beberapa tampilan aplikasi yang digunakan dalam pembuatan aplikasi ini.

4.3.1. Icon Aplikasi pada ponsel

Icon yang dilingkari dengan warna merah adalah contoh tampilan Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android setelah diinstal pada salah satu ponsel Android. Aplikasi yang di buat telah di uji dengan OS Android versi yang terendah yaitu versi 2.3 Gingerbread sampai yang tertinggi versi 4.4.2 KitKat

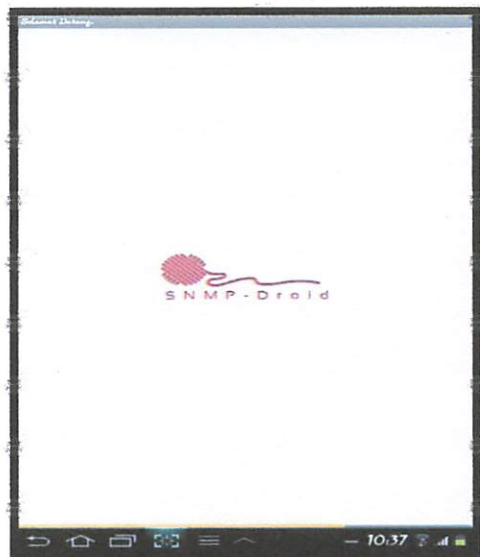


Gambar 4.1 Tampilan Icon Aplikasi

Untuk menjalan Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android ini, user cukup memilih icon yang di lingkari warna merah yang ada pada layar smartphone android seperti pada gambar 4.1..

4.3.2. Tampilan Awal Aplikasi

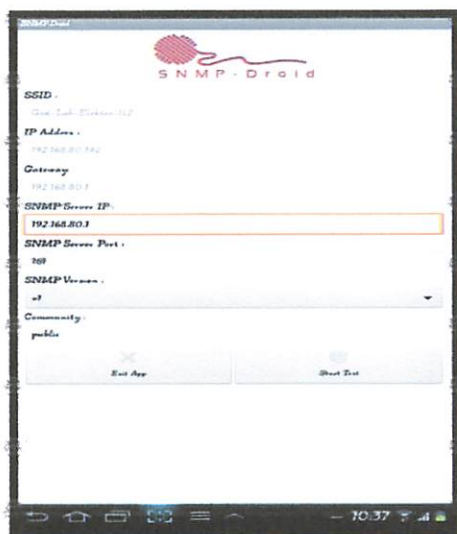
Tampilan pada gambar 4.2 merupakan tampilan awal ketika Aplikasi Mobile Remote Network Monitoring Menggunakan Mobile Device Berbasis Android di jalankan.



Gambar 4.2 Tampilan Awal aplikasi

4.3.3. Tampilan Menu Input Ip Jaringan

Tampilan pada gambar 4.3 merupakan tampilan menu input Ip jaringan yang akan di monitoring.

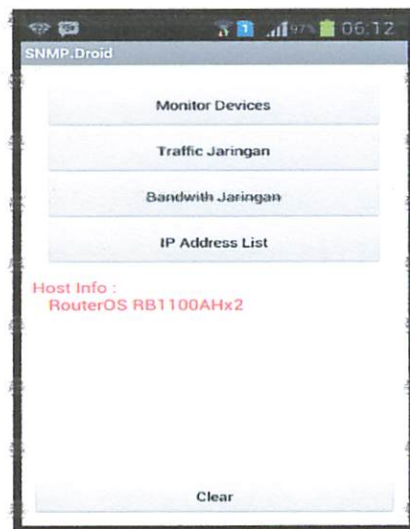


Gambar 4.3. Tampilan Menu Input IP Jaringan

Pada bagian ini, user menginput Ip jaringan yang akan dimonitoring pada kolom SNMP Server IP pada aplikasi mobile remote network monitoring menggunakan mobile device berbasis android untuk memonitoring jaringan tersebut. Percobaan dilakukan dengan memonitoring router jaringan di kampus ITN 2 dengan Ip Address jaringan 192.168.80.1.

4.3.4. Tampilan Monitoring

Tampilan pada gambar 4.4 merupakan tampilan monitoring jaringan dengan Host Info Router Os RB1100AH.2 dimana user mengaktifkan aplikasi dengan memilih button Start Test.



Gambar 4.4 Tampilan Monitoring Jaringan

Untuk melakukan proses monitoring user tinggal mengaktifkan button yang ada pada tampilan monitoring jaringan. Berikut ini fungsi button yang ada pada tampilan monitoring jaringan:

1. Button “Monitoring Devices”
Fungsinya untuk mengetahui daftar perangkat yang terhubung ke jaringan dan daftar alamat IP Address yang digunakan pada jaringan
2. Button “TrafikJaringan”
Fungsinya untuk mengetahui data input dan output pada tiap pengguna trafik jaringan

3. Button “Bandwidth Jaringan”

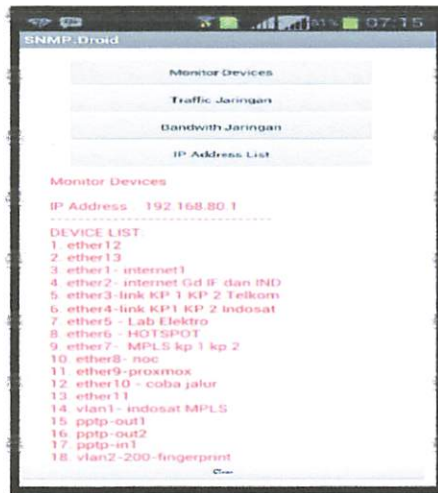
Fungsinya untuk mengetahui kecepatan upload dan download data pada tiap pengguna bandwidth jaringan

4. Button “Ip Address List”

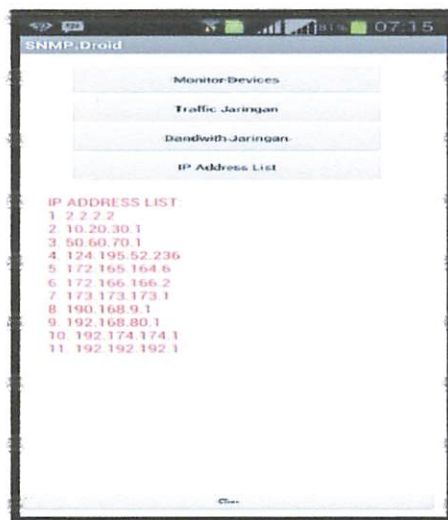
Fungsinya untuk mengetahui IP Address pengguna atau client yang sedang terhubung ke jaringan.

4.3.5. Tampilan Hasil Monitoring Devices

Tampilan pada gambar 4.5 dan gambar 4.6 merupakan tampilan monitoring devices dimana user mengaktifkan aplikasi dengan memilih button Monitor Devices.



Gambar 4.5 Tampilan Hasil Monitoring Devices

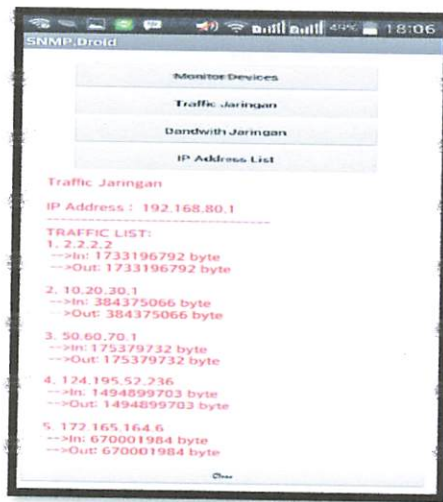


Gambar 4.6 Tampilan Hasil Monitoring Devices

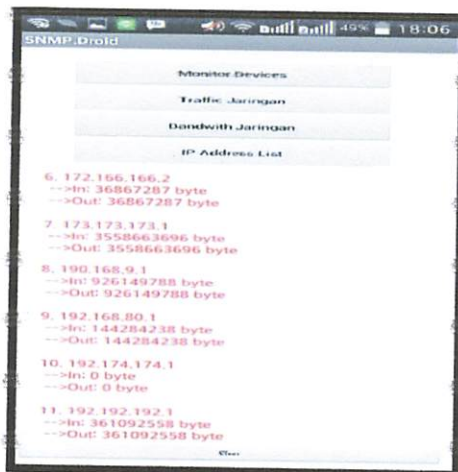
Dengan mengaktifkan button monitor devices dengan IP Address 192.168.80.1 pada aplikasi ini maka, kita dapat mengetahui daftar perangkat yang terhubung pada jaringan dan daftar alamat IP Address yang digunakan pada jaringan tersebut. Tampilan Devices List pada gambar merupakan daftar perangkat yang digunakan pada jaringan tersebut dan tampilan IP Address List merupakan daftar alamat IP Address yang digunakan.

4.3.6. Tampilan Hasil Monitoring Traffic Jaringan

Tampilan pada gambar 4.7 dan 4.8 merupakan tampilan hasil monitoring traffic jaringan dimana user mengaktifkan aplikasi dengan memilih button traffic jaringan.



Gambar 4.7 Tampilan Hasil Monitoring Traffic Jaringan

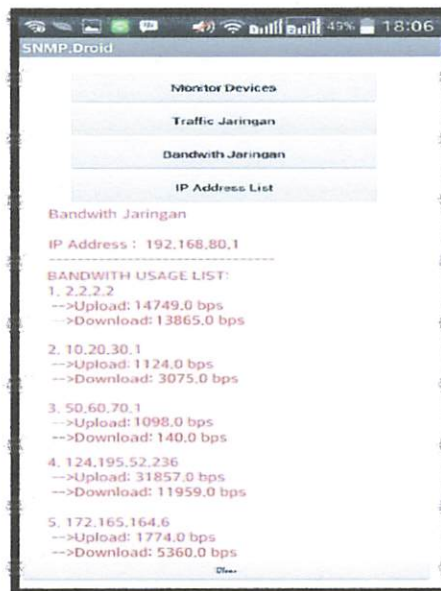


Gambar 4.8 Tampilan Hasil Monitoring Traffic Jaringan

Dengan mengaktifkan button Traffic Jaringan dengan IP Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan data input dan output pada setiap Ip Address pengguna traffic jaringan secara real time. Pada IP Address 2.2.2.2 data input output 1733196792 byte = 1.6 GB, IP Address 10.20.30.1 data input output 384375066 byte = 366.56 MB, IP Address 50.60.70.1 data input output 175379732 byte =167.2 MB, IP Address 124.195.52.236 data input output 1494899703 byte = 1.3 GB, IP Address 172.165.164.6 data input output 670001984 byte = 638.9 MB, IP Address 172.166.166.2 data input output 36867287 byte = 35.1MB, IP Address 173.173.173.1 data input output 3558663696 byte = 3.3 GB, IP Address 190.168.9.1 data input output 926149788 byte = 883.2m MB, IP Address 192.168.80.1 data input output 144284248 byte = 137.6 MB, IP Address 192.174.174.1 data input output 0 byte berarti pada IP Address ini tidak ada aktifitas jaringan, IP Address 192.192.192.1 data input output 361092558 byte = 344.3 MB

4.3.7. Tampilan Hasil Monitoring Bandwidth Jaringan

Tampilan pada gambar 4.9 dan 4.10 merupakan tampilan hasil monitoring bandwidth jaringan dimana user mengaktifkan aplikasi dengan memilih button Bandwidth jaringan.



Gambar 4.9 Tampilan Hasil Monitoring Bandwidth Jaringan

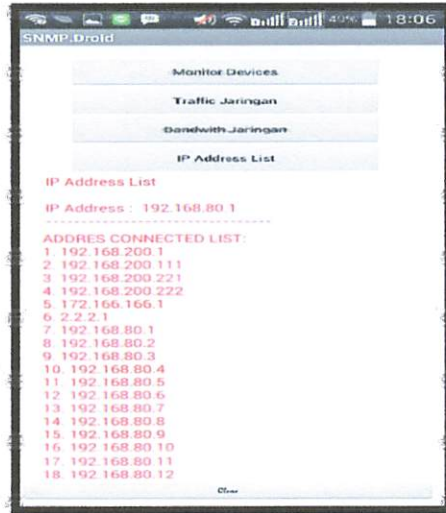


Gambar 4.10 Tampilan Hasil Monitoring Bandwidth Jaringan

Dengan mengaktifkan button Bandwidth Jaringan dengan Ip Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan kecepatan data upload dan dowload pada daftar Ip address pengguna bandwidth jaringan secara real time. Ukuran baku untuk menentukan besaran bandwith adalah bps (bit per second) 1 byte =8 bit. Pada IP 2.2.2.2 kecepatan data upload 14749.0 bps = 14.403 kbps dan data download 13865.0 bps = 13.540 kbps, pada IP 10.20.30.1 kecepatan data upload 1224.0 bps =1.195 kbps dan data download 3057.0 bps =2.98 kbps, Pada IP 50.60.70.1 data upload 1098.0 bps = 1.072 kbps dan data download 140.0 bps = 0.136 kbps, pada IP 124.195.52.236 data upload 31857.0 bps = 31.110 kbps dan data download 11959.0 bps = 11.67 kbps, IP 172.165.164.6 data upload 1774.0 bps = 1.732 kbps dan data download 5360.0 bps = 5.234 kbps ,IP 173.166.166.2 data upload 7.0 bps = 0.006 kbps dan data download 29.0 bps = 0.028 kbps ,IP 173.173.173.1 data upload 5120.0 bps = 5 kbps dan data download 28472.0 bps = 27.804 kbps ,IP 192.168.9.1 data upload 2576.0 bps = 2.515 kbps dan data download 740.0 bps = 0.722 kbps ,IP 192.168.80.1 data upload 58.0 bps = 0.056 kbps dan data download 1154.0 bps = 1.126 kbps ,IP 192.174.174.1 data upload dan download 0.0 bps berarti pada IP ini tidak ada aktifitas aliran data, IP 192.192.192.1 data upload 228.0 bps = 0.222 kbps dan data download 288.0 bps = 0.281 kbps

4.3.8. Tampilan Hasil Monitoring Ip Address List

Tampilan pada gambar 4.11 dan 4.12 merupakan tampilan hasil monitoring ip address list dimana user mengaktifkan aplikasi dengan memilih button ip address list.



Gambar 4.11 Tampilan Hasil Monitoring Ip Address List



Gambar 4.12 Tampilan Hasil Monitoring Ip Address List

Dengan mengaktifkan button Ip Address List dengan IP Address 192.168.80.1 pada aplikasi ini, maka kita dapat mengetahui IP Address pengguna atau client yang sedang terhubung ke jaringan. Gambar 4.11 dan gambar 4.12 merupakan daftar alamat Ip Address client yang sedang terhubung ke jaringan.

4.4. Tabel Hasil Pengujian

Pada percobaan kali ini penulis melakukan uji coba pada hari yang berbeda.

4.4.1 Pengujian Hari Pertama

Pada pengujian pada hari pertama dengan memonitoring jaringan di kampus 2 ITN Malang dengan hasil sebagai berikut:

1. Tabel Hasil Monitoring Devices Pada Hari Pertama

Hasil percobaan pertama memonitoring devices jaringan dapat di lihat pada tabel di bawa ini.

Tabel.4.1 Hasil Monitoring Device Pada Hari Pertama

N0	DEVICE LIST	IP ADDRESS LIST
1	Ether 12	2.2.2.2
2	Ether 13	10.20.30.1
3	Ether 1-internet 1	50.60.70.1
4	Ether 2-internet GD dan IND	124.195.52.236
5	Ether 3- link KP1 KP2 telkom	172.165.164.6
6	Ether 4-link KP1 KP2 indosat	172.166.166.2
7	Ether5-Lab Elektro	173.173.173.1
8	Ether 6-HOTSPOT	190.168.9.1
9	Ether 7-MPLS kp1 kp2	192.168.80.1
10	Ether8-noc	192.174.174.1
11	Ether9-proxmox	192.192.192.1
12	Ether 10-coba jalur	
13	Ether 11	
14	Vlan1-indosat MPLS	
15	ppth-out1	
16	ppth-out2	
17	ppth-in1	
18	vlan2-200-fingerprint	

Dengan mengaktifkan button monitor devices dengan IP Address 192.168.80.1 pada aplikasi ini maka, kita dapat mengetahui daftar perangkat yang terhubung

pada jaringan dan daftar alamat IP Address yang digunakan pada jaringan tersebut. Tampilan Devices List merupakan daftar perangkat yang digunakan pada jaringan tersebut dan tampilan IP Address List merupakan daftar alamat Ip Address yang digunakan.

2. Tabel Hasil Monitoring Traffic Jaringan Pada Hari Pertama

Hasil percobaan pertama memonitoring traffic jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.2 Hasil Monitoring Traffic Jaringan Pada Hari Pertama

N0	Traffic List	input	output
1	2.2.2.2	1733196792 byte	1733196792 byte
2	10.20.30.1	384375066 byte	384375066 byte
3	50.60.70.1	175379732 byte	175379732 byte
4	124.195.52.236	1494899703 byte	1494899703 byte
5	172.165.164.6	670001984 byte	670001984 byte
6	172.166.166.2	36867287 byte	36867287 byte
7	173.173.173.1	3558663696 byte	3558663696 byte
8	190.168.9.1	926149788 byte	926149788 byte
9	192.168.80.1	144284248 byte	144284248 byte
10	192.174.174.1	0 byte	0 byte
11	192.192.192.1	361092558 byte	361092558byte

Dengan mengaktifkan button Traffic Jaringan dengan IP Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan data input dan output pada setiap Ip Address pengguna traffic jaringan secara real time.

3. Tabel Hasil Monitoring Bandwidth Jaringan Pada Hari Pertama

Hasil percobaan Pertama memonitoring traffic jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.3 Hasil Monitoring Bandwidth Jaringan Pada Hari Pertama

N0	Bandwidth Usage List	Upload	Download
1	2.2.2.2	14749.0 bps	13865.0 bps
2	10.20.30.1	1224.0 bps	3057.0 bps
3	50.60.70.1	1098.0 bps	140.0 bps
4	124.195.52.236	31857.0 bps	11959.0 bps
5	172.165.164.6	1774.0 bps	5360.0 bps
6	173.166.166.2	7.0 bps	29.0 bps
7	173.173.173.1	5120.0 bps	740.0 bps
8	192.168.9.1	2576.0 bps	740.0 bps
9	192.168.80.1	58.0 bps	1154.0 bps
10	192.174.174.1	0.0 bps	0.0 bps
11	192.192.192.1	288.0 bps	288.0 bps

Dengan mengaktifkan button Bandwidth Jaringan dengan IP Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan kecepatan data upload dan dowload pada daftar Ip Address pengguna bandwidth jaringan secara real time

4. Tabel Hasil Monitoring Ip Address List Pada Hari Pertama

Hasil percobaan pertama memonitoring Ip Address List pada jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.4 Monitoring Ip Address Lis Pada Hari Pertama

N0	ADDRESS CONNECTED LIST
1	192.168.200.1
2	192.168.200.111
3	192.168.200.221
4	192.168.200.222
5	172.166.166.1
6	2.2.2.1
7	192.168.80.1
8	192.168.80.2
9	192.168.80.3

10	192.168.80.4
11	192.168.80.5
12	192.168.80.6
13	192.168.80.7
14	192.168.80.8
15	192.168.80.9
16	192.168.80.10
17	192.168.80.11
18	192.168.80.12
19	192.168.80.13
20	192.168.80.14
21	192.168.80.15
22	192.168.80.16
23	192.168.80.148
24	192.168.80.149
25	192.168.80.152
26	192.168.80.156
27	192.168.80.164
28	192.168.80.194
29	192.168.80.198
30	192.168.80.203
31	192.168.80.218
32	192.168.80.225
33	192.168.80.226
34	192.168.80.228
35	192.168.80.230
36	192.168.80.253
37	192.168.80.254
38	192.192.192.1
39	192.174.174.10
40	124.195.52.225
41	124.195.52.233

Dengan mengaktifkan button Ip Address List dengan IP Address 192.168.80.1 pada aplikasi ini, maka kita dapat mengetahui Ip Address pengguna atau client

yang sedang terhubung ke jaringan. Tabel 4.4 merupakan daftar alamat Ip Address client yang sedang terhubung ke jaringan.

4.4.2 Pengujian Pada Hari Kedua

Pengujian pada hari kedua dengan memonitoring jaringan di kampus 2 ITN Malang dengan hasil sebagai berikut:

1. Tabel Hasil Monitoring Devices Pada Hari Kedua

Hasil percobaan kedua memonitoring devices jaringan dapat dilihat pada tabel di bawah ini.

Tabel.4.5 Hasil Monitoring Device Pada Hari Kedua

N0	DEVICE LIST	IP ADDRESS LIST
1	Ether 12	2.2.2.2
2	Ether 13	10.20.30.1
3	Ether 1-internet 1	50.60.70.1
4	Ether 2-internet GD dan IND	124.195.52.236
5	Ether 3- link KP1 KP2 telkom	172.165.164.6
6	Ether 4-link KP1 KP2 indosat	172.166.166.2
7	Ether5-Lab Elektro	173.173.173.1
8	Ether 6-HOTSPOT	190.168.9.1
9	Ether 7-MPLS kp1 kp2	192.168.80.1
10	Ether8-noc	192.174.174.1
11	Ether9-proxmox	192.192.192.1
12	Ether 10-coba jalur	
13	Ether 11	
14	Vlan1-indosat MPLS	
15	ppth-out1	
16	ppth-out2	
17	ppth-in1	
18	vlan2-200-fingerprint	

Dengan mengaktifkan button monitor devices dengan IP Address 192.168.80.1 pada aplikasi ini maka, kita dapat mengetahui daftar perangkat yang terhubung pada jaringan dan daftar alamat IP Address yang digunakan pada jaringan tersebut. Tampilan Devices List merupakan daftar perangkat yang digunakan pada jaringan tersebut dan tampilan IP Address List merupakan daftar alamat Ip Address yang digunakan.

2. Tabel Hasil Monitoring Traffic Jaringan Pada Hari Kedua

Hasil percobaan kedua memonitoring traffic jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.6 Hasil Monitoring Traffic Jaringan Pada Hari Kedua

N0	Traffic List	input	output
1	2.2.2.2	1738860419 byte	1738860419 byte
2	10.20.30.1	384592232 byte	384592232 byte
3	50.60.70.1	2484194357 byte	2484194357 byte
4	124.195.52.236	3639393823 byte	3639393823byte
5	172.165.164.6	298153641 byte	298153641 byte
6	173.166.166.2	18683810 byte	18683810 byte
7	173.173.173.1	935776471 byte	935776471 byte
8	192.168.9.1	697938982 byte	697938982 byte
9	192.168.80.1	72882027 byte	72882027 byte
10	192.174.174.1	0 byte	0 byte
11	192.192.192.1	220904744 byte	220904744 byte

Dengan mengaktifkan button Traffic Jaringan dengan IP Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan data input dan output pada setiap

3. Tabel Hasil Monitoring Bandwidth Jaringan Pada Hari Kedua

Hasil percobaan kedua memonitoring traffic jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.7 Hasil Monitoring Bandwidth Jaringan Pada Hari Kedua

N0	Bandwidth Usage List	Upload	Download
1	2.2.2.2	15672.0 bps	13911.0 bps
2	10.20.30.1	1125.0bps	3075.0 bps
3	50.60.70.1	31641.0 bps	19873.0 bps
4	124.195.52.236	220005.0 bps	29115.0 bps
5	172.165.164.6	695.0 bps	2385.0 bps
6	173.166.166.2	3.0 bps	14.0 bps
7	173.173.173.1	4367.0 bps	7486.0 bps
8	192.168.9.7	1072.0 bps	558.0 bps
9	192.168.80.1	29.0 bps	583.0 bps
10	192.174.174.1	0.0 bps	0.0 bps
11	192.192.192.1	142.0 bps	176.0 bps

Dengan mengaktifkan button Bandwidth Jaringan dengan IP Address 192.168.80.1 pada aplikasi ini maka kita mengetahui semua lalu lintas data yang di koneksi ke jaringan dan menampilkan kecepatan data upload dan dowload pada daftar IP Address pengguna bandwidth jaringan secara real time.

4. Tabel Hasil Monitoring Ip Address List Pada Hari Kedua

Hasil percobaan kedua memonitoring Ip Address List pada jaringan dapat di lihat pada tabel di bawa ini.

Tabel 4.8 Monitoring Ip Adres Lis Pada Hari Kedua

N0	ADDRESS CONNECTED LIST
1	190.168.9.2
2	192.168.200.1
3	192.168.200.111
4	192.168.200.221
5	192.168.200.222
6	172.166.166.1

7	2.2.2.1
8	192.168.80.1
9	192.168.80.2
10	192.168.80.5
11	192.168.80.7
12	192.168.80.76
13	192.192.192.1
14	192.192.192.2
15	10.20.30.1
16	192.174.174.10
17	124.195.52.225
18	124.195.52.233

Dengan mengaktifkan button Ip Address List dengan IP Address 192.168.80.1 pada aplikasi ini, maka kita dapat mengetahui IP Address pengguna atau client yang sedang terhubung ke jaringan. Tabel 4.8 merupakan daftar alamat IP Address client yang sedang terhubung ke jaringan.

Dari hasil uji coba aplikasi monitoring jaringan yang telah dilakukan dapat disimpulkan bahwa sistem monitoring jaringan yang dibuat dapat menampilkan device pengguna, Ip address pengguna dan data trafik secara real time dan dalam sebuah jaringan Internet dilalui oleh lalu lintas data, semakin banyaknya lalu lintas data maka semakin padat dan lambat, oleh karena itu dibutuhkan monitoring untuk memantau kondisi jaringan Internet, sehingga kepadatan yang terjadi bisa diawasi.

4.3 Pengujian Hasil Monitoring Jaringan Menggunakan Aplikasi dan Pc

Pada pengujian monitoring jaringan kali ini, penulis menggunakan dua perangkat yang berbeda untuk memonitoring jaringan. Hasil pengujian yang di dapat menggunakan aplikasi skripsi akan di bandingkan dengan Pc. Perangkat yang digunakan antara lain:

1. Aplikasi skripsi
2. Pc

Tabel 4.9 Hasil Monitoring Traffic Jaringan Menggunakan Aplikasi Skripsi dan Pc

Traffic List	Aplikasi Skripsi	Pc	Selisih hasil antara Pc dan Aplikasi
	In dan Out	In dan Out	In dan Out
2.2.2.2	1733196792 byte	1733199642 byte	2850 byte
10.20.30.1	384375066 byte	384376868 byte	1802 byte
50.60.70.1	175379732 byte	175381292 byte	1560 byte
124.195.52.236	1494899703 byte	1494902061 byte	2358 byte
172.165.164.6	670001984 byte	670003838 byte	1854 byte
172.166.166.2	36867287 byte	36868671 byte	1384 byte
173.173.173.1	3558663696 byte	3558666346 byte	2650 byte
190.168.9.1	926149788 byte	926151962 byte	2174 byte
192.168.80.1	144284248 byte	144285932 byte	1684 byte
192.174.174.1	0 byte	0 byte	0 byte

Tabel 4.10 Hasil Monitoring Bandwidth Jaringan Menggunakan Aplikasi Skripsi dan Pc

Bandwidth Usage List	Aplikasi Skripsi		Pc		Selisih hasil antara Pc dan Aplikasi	
	Up	Dow	Up	Dow	Up	Dow
	2.2.2.2	14749 bps	13865 bps	15206 bps	14261 bps	457 bps
10.20.30.1	1224 bps	3057 bps	1310 bps	3200 bps	86 bps	143 bps
50.60.70.1	1098 bps	140 bps	1235 bps	182 bps	137 bps	42 bps
124.195.52.236	31857 bps	11959 bps	32527 bps	12193 bps	670 bps	234 bps
172.165.164.6	1774 bps	5360 bps	1902 bps	5606	128 bps	246 bps
172.166.166.2	1567 bps	1324 bps	1710 bps	1453 bps	143 bps	129 bps
173.173.173.1	5120 bps	740 bps	5495 bps	778 bps	375 bps	38 bps
190.168.9.1	2576 bps	740 bps	2671 bps	772 bps	95 bps	32 bps
192.168.80.1	58 bps	1154 bps	73 bps	1362 bps	15 bps	208 bps
192.174.174.1	0.0 bps	0.0 bps	0.0 bps	bps	0 bps	0 bps

Tabel 4.11 Hasil Monitoring Ip Address Lis Menggunakan Aplikasi Skripsi dan Pc

NO	Aplikasi Skripsi	Pc
	Address Connected List	Address Connected List
1	192.168.200.1	✓
2	192.168.200.111	✓
3	192.168.200.221	✓
4	192.168.200.222	✓
5	172.166.166.1	✓
6	2.2.2.1	✓
7	192.168.80.1	✓
8	192.168.80.2	✓
9	192.168.80.3	✓
10	192.168.80.4	✓
11	192.168.80.5	✓
12	192.168.80.6	✓
13	192.168.80.7	✓
14	192.168.80.8	✓

Dari hasil pengujian yang telah di lakukan, diperoleh perbedaan atau selisih hasil antara aplikasi dan pc. Pada hasil pengujian di dapatkan nilai selisi tidak lebih dari 2850 byte atau 2.7 Kb untuk trafik jarinagan dan 457 bps atau 0.05 kbps untuk bandwidth Jaringan.

Perbedaan atau selisih antara aplikasi dengan Pc, hasilnya aplikasi selalu lebih kecil jika dibandingkan Pc. Hal ini terjadi karena prinsip kerja dari aplikasi yang dibuat mengambil nilai pada saat tertentu secara periodik. Sedangkan pada Pc secara terus menerus mengambil nilai untuk diamati pada saat proses berlangsung.



BAB V

PENUTUP

Dalam bab ini akan disampaikan kesimpulan dan saran yang diperoleh dari hasil pembahasan dari BAB IV adalah sebagai berikut.

5.1. Kesimpulan

Dari hasil uji coba yang dilakukan di bab sebelumnya, diperoleh kesimpulan sebagai berikut :

1. Hasil aplikasi mobile remote network monitoring menggunakan mobile device berbasis android pada saat pengujian dapat menjalankan semua fungsi sesuai perancangan.
2. Aplikasi di uji dengan OS Android versi yang terendah yaitu versi 2.3 Gingerbread sampai yang tertinggi versi 4.4.2 KitKat.
3. Dari hasil uji coba aplikasi monitoring jaringan yang telah dilakukan dapat disimpulkan bahwa sistem monitoring jaringan yang dibuat dapat menampilkan device pengguna, Ip address pengguna dan data trafik secara real time.
4. Dari hasil percobaan di peroleh perbedaan atau selisih tidak lebih dari 2850 byte atau 2.7 KB untuk trafik jaringan dan 457 bps atau 0.05 KBps untuk bandwidth jaringan, hal ini terjadi karena prinsip kerja dari aplikasi yang dibuat mengambil nilai pada saat tertentu secara periodik. Sedangkan pada Pc secara terus menerus mengambil nilai untuk diamati pada saat proses berlangsung.

5.2 Saran

- 1 Pada kenyataannya program aplikasi yang di buat masih memiliki beberapa kekurangan dan kekurangan tersebut dapat diperbaiki dan perlu pengembangan lebih lanjut pada masa yang akan datang.
- 2 Fitur pendukung untuk monitoring jaringan dirasa masih kurang dan perlu ditambahkan lagi dan perlu pengembangan lebih lanjut.

DAFTAR PUSTAKA

- [1] Edhy Sutanta. 2005. ***Komunikasi Data dan Jaringan Komputer***. Yogyakarta: Graha Ilmu Yogyakarta.
- [2] Andri Kristanto. 2003. ***Jaringan Komputer***. Jakarta : Modul Perkuliahan Graha Ilmu.
- [3] Indras Yuda S. 2009. ***Modul Jaringan: OSI Model***.
- [4] Sofana, Iwan. 2008. ***Membangun Jaringan Komputer***. Bandung: Penerbit Informatika.
- [5] Hermawan B. 2004. ***Menguasai Java 2 dan Object Oriented Programming***. Yogyakarta: ANDI.
- [6] Meier, R. 2010. ***Professional Android 2 Application Development***. Indianapolis: Wiley Publishing, Inc
- [7] Arief S dan Aris N. 2003. ***Sistem Monitoring Traffic Packet internet Melalui SMS Berbasis Simple Network Management Protocol (SNMP)***. Surabaya: Tugas Akhir PENS.
- [8] Limpraptono, Yudi., 2010. ***Pengembangan Aplikasi Protocol SNMP Untuk Manajemen peralatan Jaringan Intranet***. Jurnal Elektro ELTEK Vol. 1.
- [9] Utami, Sri Puji. 2008. ***Perancangan Online Network Monitoring Berbasis PHP dan SNMP***. Seminar Nasional Teknologi Informasi (SNATI), Yogyakarta.

LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT.BNI(PERSERO)MALANG Kampus I :Jl. Bendungan Sigura-gura No.2 Telp. (0341)551431(Hunting), Fax (0341)553015 Malang 65145
BANK NIAGA MALANG Kampus II :Jl.Raya Karanglo, Km.2 Telp.(0341)417636 Fax (0341)417634 malang

BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama : Dominikus Duli Kadu
NIM : 09.12.504
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul Skripsi : **PERANCANGAN APLIKASI MOBILE REMOTE NETWORK
MONITORING MENGGUNAKAN MOBILE DEVICE
BERBASIS ANDROID**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 21 Februari 2015 *A*
Nilai : 78,75 (B+)

Panitia Ujian Skripsi

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.Y.1030100358

Sekretaris Majelis Penguji

Dr. Eng. I Komang Somawirata, ST, MT
NIP.Y.1030100361

Anggota Penguji,

Penguji I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Penguji II

Irmalia Faradisa, ST, MT
NIP.P.1030100365



PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil Ujian Skripsi Jurusan Teknik Elektro Jenjang Strata Satu (S-1) yang diselenggarakan pada :

Hari : Sabtu

Tanggal : 21 Februari 2015

Telah dilakukan perbaikan skripsi oleh :

Nama : Dominikus Duli Kadu

NIM : 09.12.504

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer

Judul Skripsi : PERANCANGAN APLIKASI MOBILE REMOTE NETWORK MONITORING MENGGUNAKAN MOBILE DEVICE BERBASIS ANDROID

No	Materi Perbaikan	Paraf
1	<p>Penguji I</p> <ul style="list-style-type: none">- Bahasa asing (inggris) yang tidak diterjemakan diketik miring (italic)- Program diuji dengan os android versi yang renda sampai yang tertinggi (kitkat)- Hasil pengujian	
2	<p>Penguji II</p> <ul style="list-style-type: none">- Kesimpulan dari hasil pengujian- Perancangan untuk eclips di lengkapi- Hasil pengujian.	

Disetujui,

Penguji I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Penguji II

Irmalia Faradisa, ST, MT
NIP.P.1030100365

Mengetahui,

Pembimbing I

Dr. Eng. Aryunto Seotedjo, ST, MT
NIP.P.1030800417

Pembimbing II

Bima Aulia Firmandani, ST



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : Dominikus Puli Kadu
 N I M : 0912504
 Semester : 9
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : TEKNIK ENERGI LISTRIK
TEKNIK ELEKTRONIKA
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI
 Alamat : Karangploso

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.


Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

- | | |
|--|---------|
| 1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya | (.....) |
| 2. Telah lulus dan menyerahkan laporan Praktek Kerja | (.....) |
| 3. Telah lulus seluruh mata kuliah keahlian (MKB)sesuai konsentrasinya | (.....) |
| 4. Telah menempuh matakuliah > 134 sks dengan IPK > 2 dan tidak ada nilai E | (.....) |
| 5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan | (.....) |
| 6. Memenuhi persyaratan administrasi | (.....) |

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenarannya data tersebut diatas
 Recording Teknik Elektro S-I

Malang,.....2013
 Pemohon


 (.....)






 (.....)

Ditgetahui

Mengetahui

MONITORING BIMBINGAN SKRIPSI
SEMESTER GENAP TAHUN AKADEMIK 2013-2014

Nama : Dominikus Duli Kadu
 Nim : 09.12.504
 Dosen Pembimbing : Dr.Eng.Aryuanto Seotedjo,ST.MT
 Judul : PERANCANGAN APLIKASI MOBILE REMOTE
 NETWORK MONITORING MENGGUNAKAN MOBILE DEVICVE BERBASIS
 ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
8	18/11 2014	12-00	Tabel Pengujian	
9	7/12 2014	9-30	Tabel hasil dan pengujian dijelaskan	
10	27/12/14	17-00	- Pengujian dg s/w lain	
11	11/2/15	10-30	ACC Bab <u>IV</u> <u>V</u> Seminar Hasil	
12				
13				
14				

Malang, 2014
 Pembimbing I

Dr.Eng.Aryuanto Seotedjo,ST.MT
 NIP. 1030800417



MONITORING BIMBINGAN SKRIPSI SEMESTER GENAP TAHUN AKADEMIK 2014-2015

Nama Mahasiswa : Dominikus Duli Kadu
NIM : 09.12.504
Judul Skripsi : PERANCANAGN APLIKASI MOBILE REMOTE
NETWORK MONITORING MENGGUNAKAN
MOBILE DEVICE BERBASIS ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Jumat 9/1	11:00	- Tabel hasil pengerjaan → dijabarkan	<i>[Signature]</i>
2	Rabu 10/1	10:00	- Pengerjaan	<i>[Signature]</i>
3				
4				
5				
6				
7				

Malang, 2015
Pembimbing I

Dr. Eng. Aryuato Seotedjo, ST, MT
NIP. 1030800417



MONITORING BIMBINGAN SKRIPSI SEMESTER GENAP TAHUN AKADEMIK 2014-2015

Nama Mahasiswa : Dominikus Duli Kadu
NIM : 09.12.504
Judul Skripsi : PERANCANAGN APLIKASI MOBILE REMOTE
NETWORK MONITORING MENGGUNAKAN
MOBILE DEVICE BERBASIS ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Jumat 9/1	11:00	- Tabel hasil pengerjaan → dijelaskan	<i>[Signature]</i>
2	Senin 14/1	10:00	- Pengerjaan	<i>[Signature]</i>
3				
4				
5				
6				
7				

Malang, 2015
Pembimbing I

Dr. Eng. Aryuato Seotedjo, ST, MT
NIP. 1030800417



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Malang

MONITORING BIMBINGAN SKRIPSI SEMESTER GENAP TAHUN AKADEMIK 2013-2014

Nama : Dominikus Duli Kadu
Nim : 09.12.504
Dosen Pembimbing : Bima Aulia Firmandani, ST
Judul : PERANCANGAN APLIKASI MOBILE REMOTE
NETWORK MONITORING MENGGUNAKAN MOBILE DEVICVE BERBASIS
ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	5/9 2014	10-30	Sistem Matikan Penulisan dan batasan Masalah	
2	15/9 2014	11-20	Referensi Bab II jaringan komputer	
3	17/9 2014	9-30	Analisa Sistem dan Rancangan Diagram Alir	
4	12/9 2014	8-40	Tutup Perancangan	
5	10/11 2014	12-00	Acc Bab I II III dan lanjut ke Bab IV	
6	25/11 2014	12-30	Implementasi dan Pengujian Sistem	
7	5/12 2014	11-20	Pengujian Sistem	

Malang, 2014

Pembimbing II

Bima Aulia Firmandani, ST



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Malang

MONITORING BIMBINGAN SKRIPSI SEMESTER GENAP TAHUN AKADEMIK 2013-2014

Nama : Dominikus Duli Kadu
Nim : 09.12.504
Dosen Pembimbing : Bima Aulia Firmandani, ST
Judul : PERANCANGAN APLIKASI MOBILE REMOTE
NETWORK MONITORING MENGGUNAKAN MOBILE DEVICVE BERBASIS
ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	5/9 2014	10-30	Sistem Matrik Penulisan dan batasan Masalah	
2	15/9 2014	11-20	Referensi Bab II jaringun komputer	
3	17/9 2014	9-30	Analisa Sintom dan Rancangan Diagram Alir	
4	12/9 2014	8-40	Tugas Perancangan	
5	10/11 2014	12-00	ACC Bab I II III dan lanjut ke Bab IV	
6	25/11 2014	12-30	Implementasi dan Pengujian Sintom	
7	5/12 2014	11-20	Pengujian Sistem	

Malang, 2014
Pembimbing II

Bima Aulia Firmandani, ST



MONITORING BIMBINGAN SKRIPSI SEMESTER GENAP TAHUN AKADEMIK 2013-2014

Nim : Dominikus Duli Kadu
Nama : 09.12.504
Masa Bimbingan : 1 (Satu) Semester
Judul : PERANCANGAN APLIKASI MOBILE REMOTE NETWORK MONITORING
MENGUNAKAN MOBILE DEVICE BERBASIS ANDROID

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	5/4 2004		Revisi Bab I, II, III	
2	17/4 2014		Analisa Sistem, Rancangan diagram Alir	
3	25/5 2014		Tahap Perancangan	
4	19/10 2014		ACC BAB I, II, III diteruskan dg BAB IV	
5	12/10 2014		Revisi bab II Jaringan Komputer	
6	15/10 2014		Implementasi dan Pengujian Sistem	
7	17/11 2014		Pengujian Sistem Bab IV	

Malang,2014

Pembimbing II

Bima Aulia Firmandani, ST



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

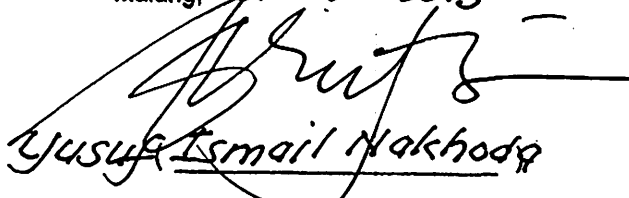
Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Dominikus Duli Kadu
NIM : 0912504
Perbaikan meliputi :

1. Bahasa asing (Inggris) yang tidak diterjemahkan di ketik miring (italic).
2. Program diuji dengan OS Android versi yang rendah sampai yang tinggi (kilkat).
3. Program aplikasi mobile diuji dibandingkan dengan jaringan yang ada di NOC.
4. Gambar atau tulisan yang mengambil dari daftar pustaka diberi indeks sesuai urutan dari daftar pustaka.
5. Hasil-hasil pengujian disimpulkan.

Malang, 21-2-2015


Yusuf Ismail Nakhoda



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Dominikus D.K
NIM : 09.12.504
Perbaikan meliputi :

Pengujian ditanyakan dibandingkan dg Ys di pe.

kesimpulan dan hasil pengujian.

Perencanaan rlt= eclipse ditanyakan.

Malang,

(Pranama)



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

nomor Surat : ITN-243/EL-FTI/2014
ampiran : -
rihal : BIMBINGAN SKRIPSI

kepada : Yth. Bapak/Ibu **Dr. Eng. Aryunto Soetedjo, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG :

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa

Nama : **DOMINIKUS DULI KADU**
Nim : **0912504**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2014-2015 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-243/EL-FTI/2014
Empiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Bima Aulia Firmandani, ST**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa

Nama : **DOMINIKUS DULI KADU**
Nim : **0912504**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2014-2015 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Rektu Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Pembuatan kode program Splash Screen

```
public class SplashScreenActivity extends Activity {
    ProgressBar pbMain;
    //tampilan utama
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        setTitle("Selamat Datang..");

        final int welcomeScreenDisplay =100;

        //loading
        pbMain = (ProgressBar) findViewById(R.id.progressBar1 );
        pbMain.setMax(welcomeScreenDisplay);

        Thread threads = new Thread() {

            int wait = 0;

            @Override
            public void run() {
                try {
                    super.run();
                    //proses loading
                    while (wait < welcomeScreenDisplay) {
                        sleep(20);
                        wait += 1;
                        //update loading
                        pbMain.setProgress(wait);
                    }
                } catch (Exception e) {
                    System.out.println("Error:" + e);
                } finally {
                    newIntens();
                }
            }
        };
        threads.start();
    }
    //tampil pesan
    public void t(String text) {
        Toast.makeText(getApplicationContext(), text, Toast.LENGTH_LONG).show();
    }
    //menjalankan tab utama
    protected void newIntens() {
        // TODO Auto-generated method stub
        startActivity(new Intent( this,ConnActivity.class ));
        finish();
    }
}
```

Untuk menjalankan SNMP Server Port masukan pemrograman sebagai berikut:

```
public class ResultActivity extends Activity {
    Utils u = new Utils();
    TextView txA;

    Button btn1,btn2,btn3,btn4;

    // file hasil
    private String fileTemp = "/sdcard/snmp.db";
    private String IP;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.res);

        Intent i = getIntent();
        IP= i.getStringExtra("IP");
        // tampilan
        txA = (TextView) findViewById(id.txtA);
        if (new File(fileTemp).exists()) {
            txA.setText( u.files.toString(fileTemp));
        }else {
            txA.setText("[No result]");
        }
        //to,bol reset
        Button btnA = (Button) findViewById(id.btnA);
        btnA.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                txA.setText("");
                u.files.Reset(fileTemp);
            }
        });

        btn1 = (Button) findViewById(id.btnMonitor);
        btn2 = (Button) findViewById(id.btnTraiff);
        btn3 = (Button) findViewById(id.btnBand);
        btn4 = (Button) findViewById(id.btn ip address);

        btn1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                showBandwith(((Button) v).getText().toString(), 1 );
            }
        });

        btn2.setOnClickListener(new View.OnClickListener() {

            @Override
```

```

        public void onClick(View v) {
            // TODO Auto-generated method stub
            showBandwith(((Button) v).getText().toString(),2 );
        }
    });
    btn3.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            showBandwith(((Button) v).getText().toString(),3 );
        }
    });
    btn4.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            showBandwith(((Button) v).getText().toString(),4 );
        }
    });
}

protected void showBandwith(String data, int string) {
    // TODO Auto-generated method stub
    StringBuffer sb = new StringBuffer();
    sb.append(data+"\n\n");
    sb.append("IP Address : "+IP+"\n");
    sb.append("-----\n");
    sb.append("\n");

    switch (string) {
    case 1:
        String ip = IP;
        String community = "public";
        String targetOid = ".1.3.6.1.2.1.2.2.1.2";//".1.3.6.1.2.1.4.20.1.1";
        sb.append("DEVICE LIST:\n");
        ArrayList<VariableBinding> ret = snmpWalk(ip, community, targetOid);
        if (ret!=null) {
            for (int j = 0; j < ret.size(); j++) {
                VariableBinding vb = ret.get(j);
                if (!"".equals(vb.getVariable().toString().trim())){
                    String[] tmpData =
vb.getVariable().toString().split(":");
                    sb.append(j+". ");
                    for (int i = 0; i < tmpData.length; i++) {
                        String strings = tmpData[i];
                        sb.append((char) Integer.parseInt(strings, 16));
                    }
                    sb.append("\n\n");
                }
            }
        }
    }
}

```

```

}else {
    sb.append("\nError get device list..");
}
sb.append("\n\n");
targetOid = ".1.3.6.1.2.1.4.20.1.1";
sb.append("IP ADDRESS LIST:\n");
ret = snmpWalk(ip, community, targetOid);
if (ret!=null) {
    for (int j = 0; j < ret.size(); j++) {
        VariableBinding vb = ret.get(j);
        if (!"".equals(vb.getVariable().toString().trim())){
            String tmpData = vb.getVariable().toString();
            sb.append((j+1)+". ");
            sb.append( tmpData);
            sb.append("\n");
        }
    }
}
}else {
    sb.append("\nError get address list..");
}
break;
case 2:
    ip = IP;
    community = "public";
    targetOid = ".1.3.6.1.2.1.2.2.1.1";///".1.3.6.1.2.1.2.2.1.2"";
    sb.append("TRAFFIC LIST:\n");
    ret = snmpWalk(ip, community, targetOid);
    targetOid = ".1.3.6.1.2.1.2.2.1.10";
    ArrayList<VariableBinding> retIN = snmpWalk(ip, community, targetOid);
    targetOid = ".1.3.6.1.2.1.2.2.1.16";
    ArrayList<VariableBinding> retOut = snmpWalk(ip, community,
targetOid);
    if (ret!=null) {
        sb.append(" error on OID .1.3.6.1.2.1.2.2.1.1 \n\n");
    }
    if (retIN!=null) {
        sb.append(" error on OID .1.3.6.1.2.1.2.2.1.10 \n\n");
    }
    if (retOut!=null) {
        sb.append(" error on OID .1.3.6.1.2.1.2.2.1.16 \n\n");
    }
    if (retIN!=null && retOut !=null && ret!= null) {
        for (int j = 0; j < ret.size(); j++) {
            VariableBinding vb = ret.get(j);
            if (!"".equals(vb.getVariable().toString().trim())){
                String[] tmpData =
vb.getVariable().toString().split(":");
                sb.append((j+1)+". ");
                for (int i = 0; i < tmpData.length; i++) {
                    String strings = tmpData[i];
                    sb.append((char) Integer.parseInt(strings, 16));
                }
            }
        }
    }
}

```



```

        double down = retSpeed.get(j).getVariable().toLong() ==
0 ? 0 :(retIN.get(j).getVariable().toLong() * 8 * 100) /
retSpeed.get(j).getVariable().toLong();
        double upl = retSpeed.get(j).getVariable().toLong() ==
0 ? 0 :(retOut.get(j).getVariable().toLong() * 8 * 100) /
retSpeed.get(j).getVariable().toLong();

        sb.append("\n -->Upload: "+upl);
        sb.append(" bps\n -->Download: "+down );
        sb.append(" bps\n\n");
    }

}

}else{
    sb.append(" Error get Bandwith usages \n\n");
}
    break;
case 4:
    ip = IP;
    community = "public";
    targetOid = ".1.3.6.1.2.1.4.22.1.3";
    targetOid = ".1.3.6.1.2.1.4.22.1.2";
    sb.append("ADDRES CONNECTED LIST:\n");
    ret = snmpWalk(ip, community, targetOid);
    if (ret!=null) {
        for (int j = 0; j < ret.size(); j++) {
            VariableBinding vb = ret.get(j);
            if (!"".equals(vb.getVariable().toString().trim())){
                String tmpData = vb.getVariable().toString();
                sb.append((j+1)+" ");
                sb.append(tmpData);
                sb.append("\n");
            }
        }
    }
}else{
    sb.append(" Error get address connected \n\n");
}
    break;

default:
    break;
}

txA.setText(sb.toString());

}
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    if (new File(fileTemp).exists()) {
        txA.setText( u.files.toString(fileTemp));
    }else {
        txA.setText("[No result]");
    }
}

```

```

}

    private static int version = SnmpConstants.version1;
    private static String protocol = "udp";
    private static int port = 161;

    private ArrayList<VariableBinding> snmpWalk(String ip, String community,
String targetOid) {
        String address = protocol + ":" + ip + "/" + port;

        OID targetOID = new OID(targetOid);

        PDU requestPDU = new PDU();
        requestPDU.setType(PDU.GETNEXT);
        requestPDU.add(new VariableBinding(targetOID));

        CommunityTarget target = SnmpUtil.createCommunityTarget(address,
            community, version, 2 * 1000L, 3);
        TransportMapping transport = null;
        Snmp snmp = null;
        ArrayList<VariableBinding> ret= new ArrayList<VariableBinding>();
        try {
            transport = new DefaultUdpTransportMapping();
            snmp = new Snmp(transport);
            transport.listen();
            boolean finished = false;
            while (!finished) {
                VariableBinding vb = null;
                ResponseEvent response = snmp.send(requestPDU, target);
                PDU responsePDU = response.getResponse();

                if (null == responsePDU) {
                    System.out.println("responsePDU == null");
                    finished = true;
                    break;
                } else {
                    vb = responsePDU.get(0);
                }
                // check finish
                finished =checkWalkFinished(targetOID,responsePDU, vb);
                if (!finished) {
                    ret.add(vb);
                    // Dump response.

                    System.out.println("----"+vb.toString());
                    System.out.println("----"+vb.getOid().toString());
                    System.out.println("----"+vb.getVariable().toString());
                }
                if (!"".equals(vb.getVariable().toString().trim())){
                String[] tmpData = vb.getVariable().toString().split(":");
                for (int i = 0; i < tmpData.length; i++) {
                String string = tmpData[i];
                //
                //
                //
                System.out.print((char) Integer.parseInt(string, 16) );
                //
                //
                //
                System.out.println();
            }
        }
    }
}

```

```

//
//
        }
        // Set up the variable binding for the next entry.
        requestPDU.setRequestID(new Integer32(0));
        requestPDU.set(0, vb);
    }
}
System.out.println("success finish snmp walk!");
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("SNMP walk Exception: " + e);
} finally {
    if (snmp != null) {
        try {
            snmp.close();
        } catch (IOException ex1) {
            snmp = null;
        }
    }
    if (transport != null) {
        try {
            transport.close();
        } catch (IOException ex2) {
            transport = null;
        }
    }
}
}
return ret;
}
}

/**
 * check snmp walk finish
 * @param requestPDU
 * @param targetOID
 * @param responsePDU
 * @param vb
 * @return
 */
private boolean checkWalkFinished(OID targetOID, PDU responsePDU,
VariableBinding vb) {
    boolean finished = false;
    if (responsePDU.getErrorStatus() != 0) {
        System.out.println("responsePDU.getErrorStatus() != 0 ");
        System.out.println(responsePDU.getErrorStatusText());
        finished = true;
    } else if (vb.getOid() == null) {
        System.out.println("vb.getOid() == null");
        finished = true;
    } else if (vb.getOid().size() < targetOID.size()) {
        System.out.println("vb.getOid().size() < targetOID.size()");
        finished = true;
    } else if (targetOID.leftMostCompare(targetOID.size(), vb.getOid()) !=
0) {
        System.out.println("targetOID.leftMostCompare() != 0");
        finished = true;
    }
}

```

```
    } else if (Null.isExceptionSyntax(vb.getVariable().getSyntax())) {
System.out
println("Null.isExceptionSyntax(vb.getVariable().getSyntax())");
        finished = true;
    } else if (vb.getOid().compareTo(targetOID) <= 0) {
        System.out.println("Variable received is not "
+ "lexicographic successor of requested " + "one:");
        System.out.println(vb.toString() + " <= " + targetOID);
        finished = true;
    }
    return finished;
}
}
```