

MODUL TEKNOLOGI WEB SERVICE



Di susun oleh

Syarah Seimahaira, M.Kom

**Fakultas Teknologi Informasi Program Studi Sains Data
Universitas Nusa Mandiri
2022**

KONSEP INTEROPERABILITAS, SISTEM TERDISTRIBUSI

Jaringan Komputer

Dalam membangun jaringan komputer yang baik pada sebuah institusi atau perusahaan, perlu mempunyai kemampuan yang dapat mendukung jenis-jenis komponen jaringan seperti perangkat keras (*hardware*) maupun perangkat lunak (*software*) dari berbagai vendor. Dimana jaringan komputer ini:

1. Mampu menangani kendala heterogenitas dan *interoperabilitas*.
2. Memerlukan acuan (*referensi*) bagi jaringan dan arsitektur jaringan (*network architectures*) yang memenuhi suatu persyaratan spesifikasi tertentu.

Jaringan Komputer dapat mendukung konsep-konsep:

1. Open System
2. Scalability
3. Connectivity/interconnectivity
4. Interoperability
5. Ease of Implementation
6. Ease of Modification
7. Reliability

Interoperability

Suatu metode yang digunakan agar data dapat saling dikenali (*undstable*) antar computer (berlawanan dengan proprietary) atau dikutip dari Kominfo Interoperabilitas adalah kapabilitas dari suatu produk atau sistem –yang antarmukanya diungkapkan sepenuhnya –untuk berinteraksi dan berfungsi dengan produk atau sistem lain, baik saat ini ataupun di masa mendatang, tanpa batasan akses atau implementasi (*Interoperability Working Group*). Misal: perbedaan sistem operasi maupun bahasa-bahasanya.

Berdasarkan definisi tersebut, fokus interoperabilitas adalah kapabilitas sebuah sistem TIK untuk dapat berinteraksi dengan sistem lainnya. Terkait dengan kasus pada paragraf sebelumnya, adanya konsep interoperabilitas akan memungkinkan sistem yang sedang dikembangkan untuk menggunakan data yang dimiliki oleh sistem yang sudah ada, baik data dari sistem yang dimiliki oleh instansi yang sama, maupun instansi yang berbeda. Secara teknologi Interoperabilitas sendiri sebenarnya telah dikenal secara luas di Indonesia, namun karena bersifat *Back Engine* Maka *End User* tidak begitu aware akan adanya Interoperabilitas ini. Di dunia perbankan, pertukaran antarsistem Informasi ditunjukkan dalam

bentuk pertukaran dana antar bank melalui Sistem Informasi perbankan dan atau melalui ATM, hal yang sangat umum dilakukan masyarakat sehari-hari. Sedangkan di Web, Interoperabilitas telah muncul dalam bentuk yang lebih kompleks, yaitu *cloud computing*.

Contoh lainnya yaitu Google API, kita dengan mudah memasukkan posisi pada Google map dan menampikannya pada web kita, dimana ini adalah salah satu contoh interoperabilitas. Namun sedemikian hebatnya penetrasi teknologi pertukaran data antar sistem informasi ini, amat disayangkan bahwa ternyata Sistem Informasi yang dikembangkan oleh pemerintahan di Indonesia sebagian besar sangat tidak memperhatikan aspek Interoperabilitas ini. Selama ini kebanyakan Sistem Informasi yang dikembangkan oleh instansi pemerintahan hanya dapat memberikan manfaat secara lokal, terutama bagi satu pemilik anggaran pengembangan Sistem Informasi tersebut. Sebagai akibatnya, Informasi dasar yang ada pada suatu Sistem Informasi (contoh : Informasi Kependudukan) seringkali menjadi redundan terhadap Sistem Informasi lain dan tidak sinkron.

Akibat lainnya adalah sulitnya melakukan pertukaran data yang harus melalui proses pengkopian dan penyesuaian data yang panjang dan memakan waktu. Hal ini membuat Tata Sistem Informasi Kepemerintahan di Indonesia carut marut dengan pulau-pulau sistem informasi yang tersebar dimana-mana. Meskipun demikian, bukan tidak ada Sistem Informasi Kepemerintahan yang ternyata mendukung Interoperabilitas dengan baik. Ambil contoh program National Single Window (NSW) yang dicanangkan pemerintah, datanya mengambil dari Departemen Perindustrian, Bea cukai, Dirjen Postel dan beberapa Instansi lainnya.

Salah satu bentuk dari interoperability adalah interoperability government. Interoperabilitas e-Government (pemerintahan secara elektronik) saat ini sudah menjadi salah satu kebutuhan utama dalam sistem pemerintahan elektronik di Indonesia. Interoperabilitas mutlak diperlukan agar berbagai layanan pemerintah yang terhubung ke jaringan (atau yang lebih dikenal sebagai e-Service) dapat berkomunikasi secara efektif satu sama lain. Adanya komunikasi efektif tentu akan memberikan banyak manfaat bagi seluruh pihak, baik bagi pemerintah maupun masyarakat umum.

Ada banyak manfaat yang dapat diberikan dari interoperabilitas khususnya pemerintahan, diantaranya adalah:

1. Meningkatkan efisiensi kinerja aparat dan layanan pemerintahan.
2. Meningkatkan transparansi pemerintahan.
3. Membantu pemerintah dalam menentukan kebijakan.
4. Memudahkan pencarian informasi, baik untuk pemerintah maupun masyarakat.

Hal ini tidak terlepas dari perkembangan berbagai macam aplikasi Teknologi Informasi dan Komunikasi (TIK) saat ini sudah merambah dalam berbagai sisi kehidupan. Hal ini terlihat dari semakin banyaknya pengguna dan jenis alat-alat elektronik, seperti laptop, komputer, tablet, dan pad(atau yang sering dikenal dengan nama lain tab). Perkembangan dan pemanfaatan TIK, khususnya di Indonesia, tidak hanya terlihat pada kalangan industri dan swasta, namun juga dalam lingkungan pemerintahan. Saat ini, sudah terdapat beberapa jenis aplikasi TIK yang digunakan untuk memenuhi dan membantu berbagai kegiatan pemerintahan. Bahkan, hampir seluruh tingkat pemerintahan memiliki dan membangun aplikasi mereka masing-masing, baik pemerintah di tingkat propinsi maupun kotamadya. Tentu kecenderungan yang sama juga terlihat pada lembaga pemerintahan di tingkat pusat, seperti kementerian dan badan keamanan nasional. Seluruh penerapan TIK ini, tidak lain dan tidak bukan, ditujukan untuk membantu terwujudnya layanan dan pemerintahan yang baik, bersih, cepat,serta bersahabat.

Saat ini sudah ada beberapa layanan pemerintah berbasis TIK yang sangat terasa manfaatnya bagi masyarakat luas. Beberapa diantaranya adalah pendaftaran paspor yang tersambung ke jaringan (atau yang lebih dikenal dengan istilah online), publikasi hibah dari Dikti pada situs web (atau yang lebih sering dikenal dengan nama website) resmi Dikti, dan pengumuman penerimaan siswa baru untuk tingkat Sekolah Menengah Pertama (SMP) dan Sekolah Menengah Atas (SMA) yang tersambung ke jaringan. Tersedianya layanan-layanan seperti ini dapat membangun hubungan baik antara pemerintah dengan masyarakat. Sayangnya, sebagian, atau hampir seluruh aplikasi-aplikasi, pemerintahan tersebut tidak dapat berbicara dan berbagi data dengan aplikasi lainnya. Padahal, sering kali ada dua atau lebih lembaga (baik milik pemerintah ataupun yang bukan) membutuhkan data yang sama. Selain itu, tidak jarang suatu lembaga membutuhkan data yang dimiliki lembaga lainnya, contohnya data kependudukan dan identitas penduduk yang dimiliki oleh Dinas Kependudukan dan Catatan Sipil. Data ini tentu dibutuhkan oleh seluruh lembaga pemerintah lainnya, khususnya untuk memeriksa kebenaran status kewarganegaraan serta data pribadi lainnya. Apabila sistem-sistem yang membutuhkan data tersebut dapat berkomunikasi secara otomatis dengan sistem yang dimiliki Dinas Kependudukan dan Catatan Sipil, maka akan ada banyak proses administratif yang dapat disederhanakan. Tentu saja hal tersebut akan

secara langsung dapat mempermudah dan mempercepat pekerjaan pegawai pemerintah dan juga masyarakat yang menggunakan layanan tersebut. Ketika bicara mengenai TIK, bukanlah hal yang tidak mungkin untuk membuat dua sistem TIK saling berbicara. Tentu saja perlu ada kesepakatan dan kesepakatan antara dua sistem tersebut, baik dalam hal bahasa, data, maupun

keamanan. Buku yang sedang Anda baca saat ini dapat membantu mencapai cita-cita tersebut. Buku ini akan membantu Anda dalam menyusun, mendesain, dan mengembangkan aplikasi kegiatan pemerintahan yang dapat berbicara dengan aplikasi pemerintahan lainnya, yang tentu saja, tanpa melupakan asas keamanan dan kerahasiaan.

Ketika kata interoperabilitas digunakan dalam buku ini, maka secara tidak langsung istilah e-Government juga harus terbawa, interoperabilitas e-Government. Istilah e-Government sendiri merupakan kependekan dari electronic-Government, yang dapat diterjemahkan secara harfiah menjadi Pemerintahan (secara) Elektronik. Istilah ini selalu diasosiasikan sebagai pemanfaatan TIK secara maksimal untuk meningkatkan efektifitas, kinerja, dan pelayanan pemerintah bagi masyarakat umum. Saat ini di Indonesia, e-Government bukanlah suatu hal yang baru.

Beberapa contoh aplikasi layanan pemerintahan:

1. Registration

Sistem untuk pendaftaran wajib pajak secara online. Melalui sistem ini masyarakat dapat mengetahui persyaratan yang diperlukan untuk menjadi wajib pajak dan melakukan pendaftaran secara online. Kemudahan akses informasi dan pendaftaran seperti ini membuat proses registrasi secara offline (langsung atau tatap muka) menjadi lebih cepat dan memperkecil peluang kesalahan yang mungkin dilakukan Pengguna. Website: <http://ereg.pajak.go.id>

2. E-KTP

KTP Electronic-KTP (e-KTP) hak dari seluruh warga negara Indonesia. Pada e-KTP terdapat Nomor Induk Kependudukan (NIK), yaitu identitas tunggal setiap penduduk dan berlaku seumur hidup. Data kependudukan ini disatukan dalam database kependudukan nasional. e-KTP diharapkan dapat menjadi identitas tunggal, tidak dapat dipalsukan/digandakan, dan dapat digunakan sebagai kartu suara pada Pemilu atau Pilkada. Website: <http://www.e-ktp.com/>

3. E-Procurement

Aplikasi TIK pengadaan secara elektronik yang dikembangkan oleh LKPP (Lembaga Kebijakan

Pengadaan Barang/Jasa Pemerintah) merupakan contoh dari aplikasi jenis ini. LKPP menggunakan aplikasi tersebut untuk e-Tendering, e-Purchasing, dan e-Audit. Website: <http://www.lkpp.go.id/v2/content.php?mid=8474545499>

4. KPK Whistle

Fasilitas dari KPK dalam rangka menarik partisipasi masyarakat untuk turut memberantas korupsi di lingkungannya. Pada sistem ini, seseorang dapat melaporkan tindak korupsi yang ia ketahui. Website: <http://kws.kpk.go.id/>

5. SISNI

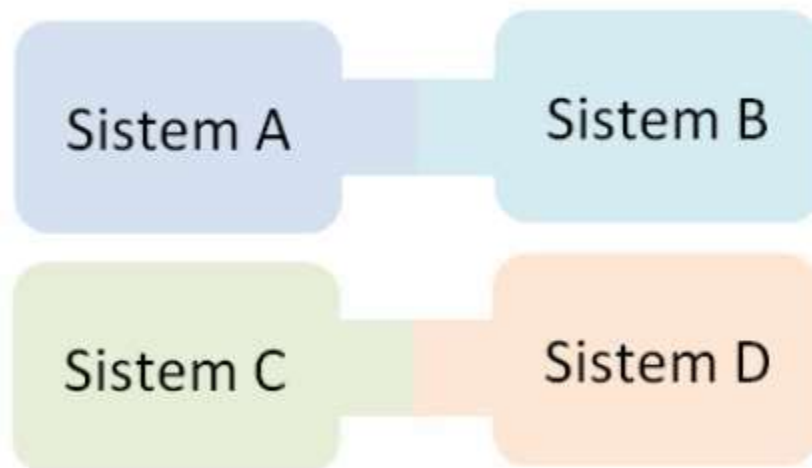
Sistem ini dikembangkan untuk membantu pembuatan Standar Nasional Indonesia (SNI) dan juga memfasilitasi e-balloting. Sistem ini memungkinkan para penggunanya untuk mengirimkan dokumen elektronik dan memberikan komentar terhadap dokumen tertentu secara elektronik. Website: <http://websisni.bsn.go.id/>

Dari contoh diatas terlihat jelas bahwa selain dikembangkan untuk tujuan yang berbeda-beda, aplikasi-aplikasi tersebut juga dikembangkan oleh instansi pemerintah yang berbeda. Artinya, data yang tersimpan dan digunakan oleh aplikasi tersebut sangat beraneka ragam. Umumnya, hanya instansi pengembang aplikasi yang akan mendapatkan hak eksklusif terhadap data tersebut. Hal ini tentu sangat dapat dipahami, mengingat besarnya biaya dan waktu yang dihabiskan untuk membangun dan mengumpulkan seluruh data yang dibutuhkan aplikasi tersebut. Namun, tidak jarang juga diantara aplikasi-aplikasi tersebut ada yang membutuhkan data yang sudah dimiliki oleh aplikasi lainnya. Sayangnya, sang pengembang aplikasi tidak tahu bahwa data yang ia butuhkan sebenarnya sudah tersedia. Pada kondisi seperti ini dapat dipastikan para pengembang akan memilih untuk mengumpulkan data dari awal lagi. Keputusan ini secara langsung akan mempengaruhi biaya pengembangan aplikasi. Tentu saja, dilihat dari sisi apapun, hal tersebut merupakan penyalahgunaan sumber daya.

1. Konsep Kompatibilitas

Membicarakan konsep kompatibilitas maka yang dibicarakan adalah kecocokan antara satu sistem dengan sistem lainnya. Kecocokan disini dapat saja berfokus pada struktur dan jenis data,

bahasa pemrograman yang digunakan, atau hal-hal lain yang disepakati para pengembang sistem. Pada Gambar dibawah terlihat bahwa Sistem C dapat berkomunikasi dengan Sistem D dan Sistem A dapat berkomunikasi dengan Sistem B. Namun, antara Sistem A dengan Sistem C atau Sistem A dengan Sistem D tidak dapat berkomunikasi. Kompatibilitas tidak dapat menjamin bahwa setiap sistem dapat memanfaatkan sumber daya yang dimiliki sistem lainnya.

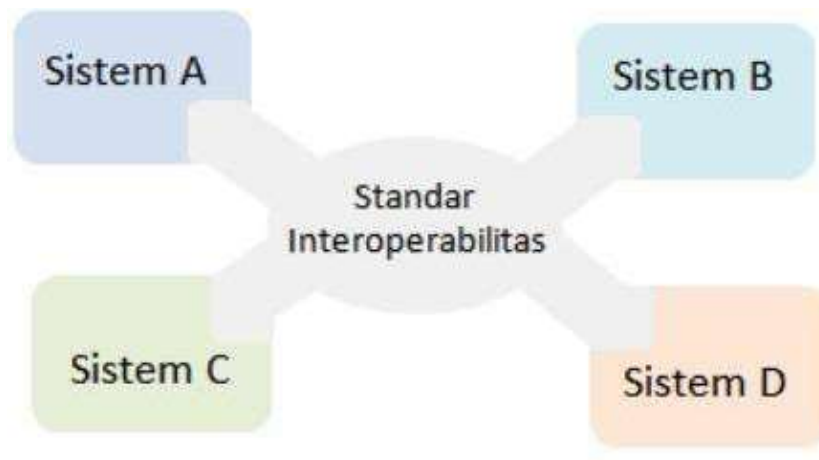


Sumber: KOMINFO, Strategi Integrasi Data Melalui Konsep Interoperabilitas Sistem Elektronik, 2017

2. Konsep Interoperabilitas

Setiap sistem diharapkan dapat memanfaatkan data milik sistem lain secara optimal. Jika Anda perhatikan, pada Gambar dibawah ini terlihat bahwa yang memungkinkan terjadinya hal tersebut adalah standar interoperabilitas. Perhatikan juga bahwa baik sistem A, B, C, maupun D memasukan standar interoperabilitas sebagai bagian dari sistem. Hal ini sangat berbeda dengan

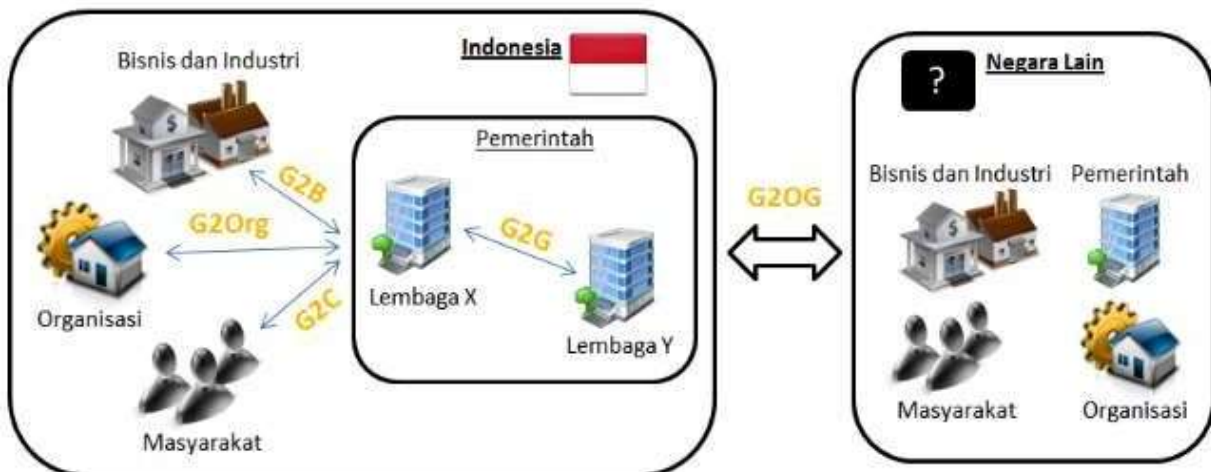
konsep kompatibilitas. Tentu saja, penggunaan teknik interoperabilitas akan lebih mudah diimplementasikan, terutama untuk pengembangan sistem-sistem TIK baru lainnya.



Sumber: Direktorat E-Government Ditjen. Aplikasi Informatika Kementerian Komunikasi Dan Informatika, 2017

Fokus Pengembangan Interoperabilitas di pemerintahan, yaitu:

1. G2B
2. G2Org
3. G2C
4. G2G
5. G2OG



Sumber: KOMINFO, Strategi Integrasi Data Melalui Konsep Interoperabilitas Sistem Elektronik, 2017

Teknik interoperabilitas akan memberikan efek positif dalam jangka panjang. Interoperabilitas memungkinkan suatu sistem untuk memanfaatkan data yang dimiliki oleh sistem-sistem lainnya. Pada panduan yang dikeluarkan oleh United Nations Development Programme (UNDP) (United Nations Development Programme, 2007) disebutkan paling tidak ada lima fokus pengembangan interoperabilitas sistem pemerintahan, yaitu:

1. Government-to-Government (G2G)

Pengembangan interoperabilitas G2G berfokus pada membangun jalur komunikasi digital antara sistem pemerintahan dengan sistem pemerintahan lainnya. Hal ini termasuk sistem pemerintahan yang berasal dari lembaga yang sama, berbeda lembaga, atau antara pemerintah pusat dengan pemerintah daerah.

2. Government-to-Business (G2B)

Pengembangan interoperabilitas ini berfokus pada membangun jalur komunikasi digital antara sistem pemerintahan dengan sistem yang dikembangkan oleh/berhubungan dengan dunia bisnis dan industri. Pengembangan difokuskan untuk mempermudah dan meningkatkan kualitas layanan pemerintah untuk kalangan bisnis dan industri.

3. Government-to-Citizens (G2C)

Pengembangan interoperabilitas ini berfokus pada pembangunan jalur komunikasi digital antara sistem pemerintahan dengan sistem yang dibangun oleh masyarakat, atau antara sistem pemerintahan dengan sistem pemerintahan lainnya guna memberikan layanan terbaik kepada masyarakat.

4. Government-to-Organizations (G2Org)

Pengembangan interoperabilitas ini berfokus pada pembangunan jalur komunikasi digital antara sistem pemerintahan dengan sistem yang dikembangkan oleh organisasi non-pemerintah.

5. Government-to-Other-Governments (G2OG)

Pengembangan interoperabilitas ini berfokus pada pembangunan jalur komunikasi digital antara sistem pemerintahan di suatu negara dengan sistem pemerintahan di negara lain. Fokus interoperabilitas seperti ini sudah diterapkan pada negara-negara di Eropa melalui European Interoperability Strategy dan European Interoperability Framework.

Sistem Terdistribusi

Defenisi system terdistribusi:

- a. Sebuah sistem yg komponennya berada pada jaringan komputer. Komponen tersebut saling berkomunikasi dan melakukan koordinasi hanya dgn pengiriman pesan (*message passing*).
- b. Sebuah sistem terdistribusi (DS) terdiri dari sekumpulan komputer yang mandiri yang terhubung dalam jaringan komputer dan dilengkapi dengan software sistem terdistribusi (*Colouris 1994*)
- c. Software Sistem Terdistribusi dapat mengkoordinir aktifitas masing-masing komputer dan melakukan pertukaran sumber daya sistem HW, SW dan data.

Sistem Terdistribusi dibutuhkan dengan beberapa alasan yaitu:

1. Performance

Sekumpulan prosesor dapat menyediakan kinerja yang lebih tinggi daripada komputer yang terpusat

2. Distribution

Banyak aplikasi yang terlibat, sehingga lebih baik jika dipisah dalam mesin yang berbeda (contoh: aplikasi perbankan, komersial)

3. Reliability

Jika terjadi kerusakan pada salah satu mesin, tidak akan mempengaruhi kinerja system secara keseluruhan

4. Incremental Growth

Mesin baru dapat ditambahkan jika kebutuhan proses meningkat

5. Sharing Data/Resource

Resource adalah:

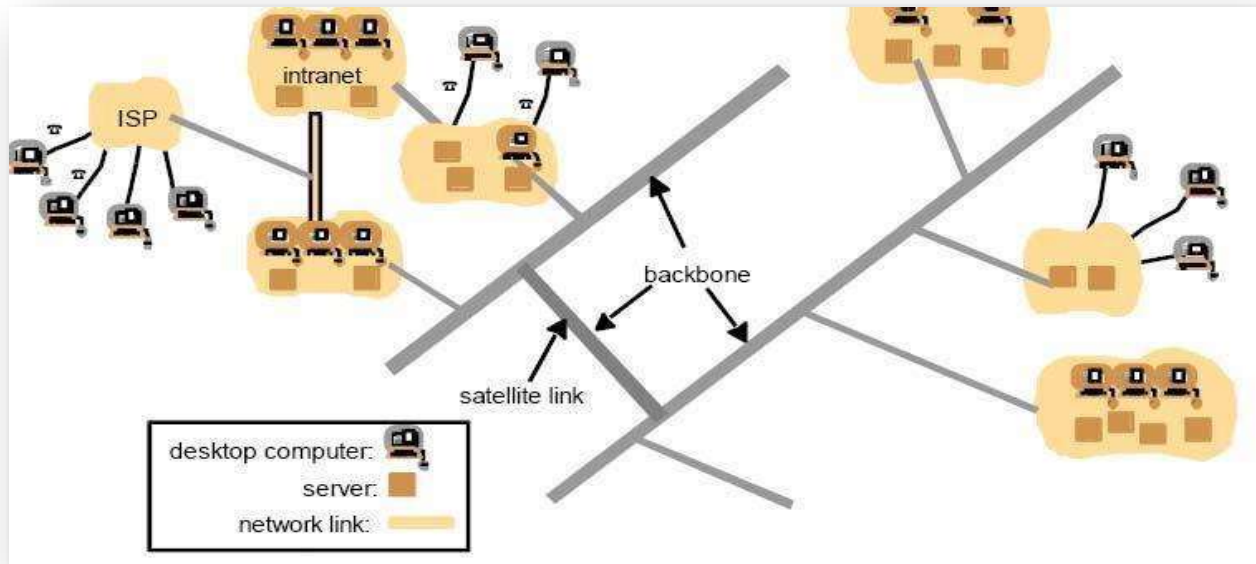
- Segala hal yang dapat digunakan bersama dalam jaringan komputer.
- Meliputi hardware (e.g. disk, printer, scanner), juga software (berkas, basis data, obyek data).

6. Communication

Menyediakan fasilitas komunikasi antar manusia

Contoh sistem terdistribusi:

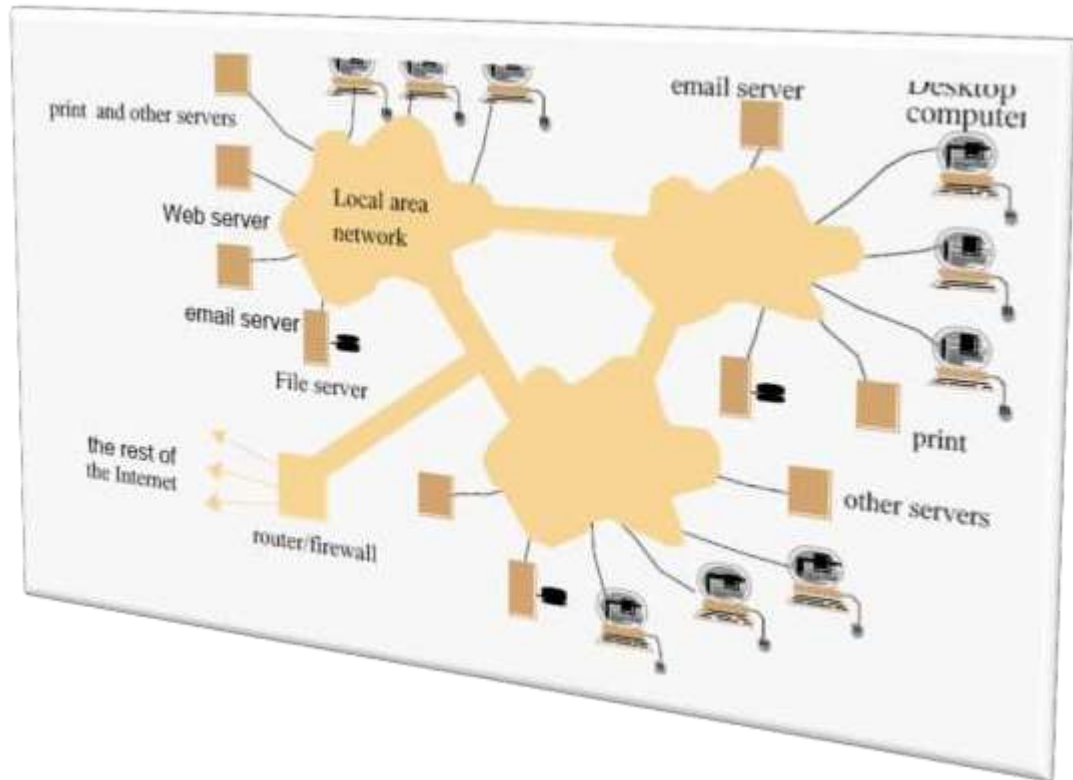
Internet, merupakan suatu bentuk jaringan global yang menghubungkan komputer dengan satu sama lainnya, yang dapat berkomunikasi dengan media IP sebagai protokol.



Sumber: Budi Susanto, Pengantar Sistem Terdistribusi

Intranet

- Jaringan yang teradministrasi secara lokal
- Biasanya proprietary
- Terhubung ke internet (melalui firewall)
- Menyediakan layanan internal dan eksternal



Sumber: Budi Susanto, Pengantar Sistem Terdistribusi

Sistem terdistribusi multimedia.

Biasanya digunakan pada infrastruktur internet

Karakteristik

Sumber data yang heterogen dan memerlukan sinkronisasi secara real time

Video, audio, text Multicast

Contoh:

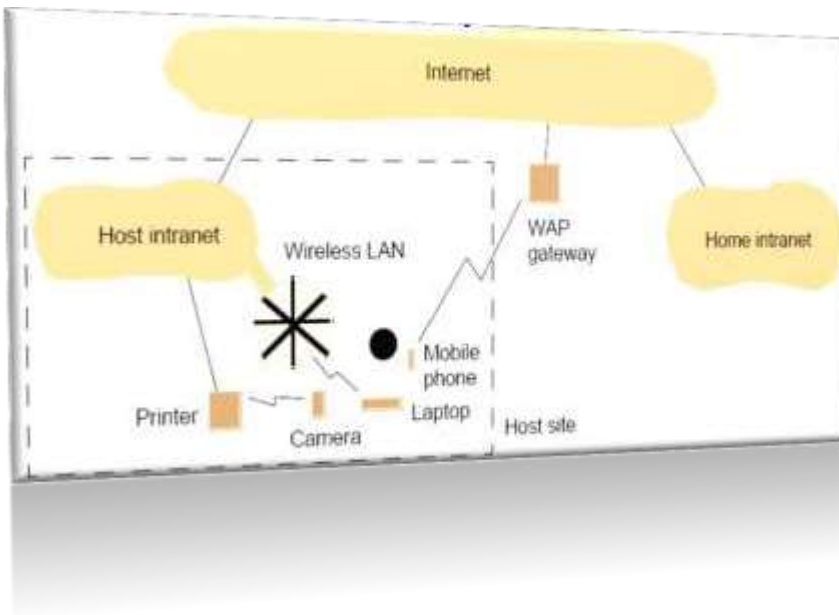
- Teleteaching tools, Video-conferencing, Video and audio on demand

Mobile dan sistem komputasi ubiquitous

Sistem telepon Cellular (e.g., GSM)

Resources dishare : frekuensi radio, waktu transmisi dalam satu frekuensi, bergerak. Komputer laptop, ubiquitous computing.

Handheld devices, PDA, etc

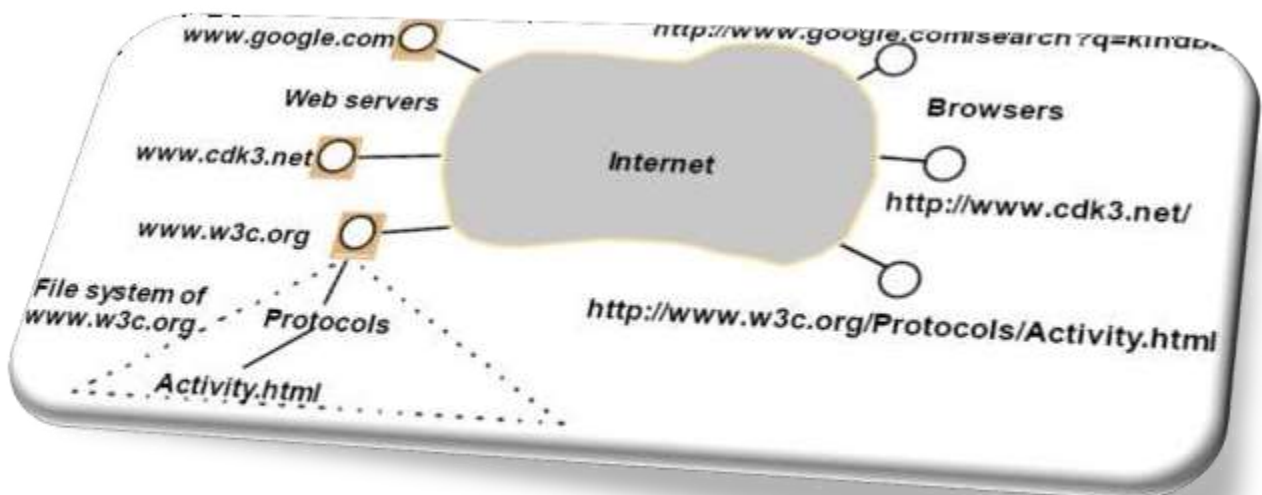


Sumber: Budi Susanto, Pengantar Sistem Terdistribusi

5. World wide web

Arsitektur client/server terbuka yang diterapkan di atas infrastruktur internet

Shared resources (melalui URL)



Sumber: Budi Susanto, Pengantar Sistem Terdistribusi

PENGENALAN WEB SERVICE

Web Service

Web Service adalah Sebuah aplikasi lintas platform yang dapat diakses melalui jaringan (*intranet* dan *internet*) dimana dalam aplikasi tersebut menyediakan berbagai fungsi dengan tujuan digunakan untuk interaksi aplikasi satu dengan aplikasi yang lain. Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah database ditanamkan, dibuat dalam bahasa apa, sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. Web service mampu menunjang interoperabilitas. Sehingga web service mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada. Web pada umumnya digunakan untuk melakukan respon dan request yang dilakukan antara client dan server.

Contoh Kasus 1 :

Implementasi Web Service adalah sistem login seperti yang ada di **Kaskus**, atau **Detik.com**. Jika Anda mengunjungi **Kaskus**, maka Anda akan menjumpai **sistem login** yang dapat menggunakan **akun Facebook, Yahoo** maupun **Twitter**. Dengan kata lain, Anda dapat bergabung dalam komunitas forum Kaskus atau istilah kerennya Kaskuser hanya dengan memiliki akun FB, Yahoo atau Twitter tanpa harus registrasi di dalam Kaskus nya sendiri.

Contoh Kasus 2 :

Contoh implementasi Web Service adalah di bagian komentar Detik.com. Setiap kali kita akan mengisi komentar di Detik.com, kita bisa menggunakan akun FB kita. Dengan penerapan teknologi ini, FB, Yahoo dan Twitter menyediakan service yang memungkinkan sistem lain menggunakan akun mereka untuk login. Dan... kebetulan Kaskus dan Detik.com ini memanfaatkan service tersebut. Dalam kasus ini, FB, Yahoo, dan Twitter dikatakan bertindak sebagai server sedangkan Kaskus dan Detik.com bertindak sebagai client.

Lihat gambar di bawah ini.



Contoh Kasus 3 :

Pengguna perangkat ponsel Android, yang sudah pernah menginstall aplikasi semacam detik.com, kompas.com atau vivanews.com, dimana Anda bisa melihat berita dari situs detik.com di ponsel Android Anda.

Pertanyaan : Apakah portal-portal tersebut dalam membuat aplikasi portal versi Android juga membuat database/data beritanya? Apakah mereka melakukan pengimputan data berita dua kali ? yaitu versi web dan versi android ?

Jawaban : sudah pasti tidak, karena mereka pasti telah menggunakan layanan web service untuk pertukaran data pada dua versi aplikasi yang mereka buat.

Jadi secara singkat **Web Service** adalah aplikasi yang dibuat agar dapat dipanggil



atau diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan. Adapun yang biasa dipakai adalah format pertukaran data dengan **XML** dan **JSON**.

Lalu, bagaimana caranya memanggil atau memanfaatkan sebuah Web Service ? sebuah web service dapat dipanggil oleh aplikasi lain dengan menggunakan bantuan HTTP (Hypertext Transfer Protocol). Web service juga memungkinkan untuk dipanggil dengan menggunakan protocol lain seperti SMTP (*Simple Mail Transfer Protocol*), namun yang paling umum digunakan HTTP.

Karena web service menggunakan protokol **HTTP**, tentu **PHP** sebagai bahasa pemrograman web menjadi salah satu kekuatan dalam bahasa pemrograman yang mengelola web service. Meski banyak teknik dan metode untuk menghasilkan web service dengan PHP.

JENIS-JENIS WEB SERVICE BERDASARKAN MANFAAT

- **Reusable application-components**

Jenis web service yang dimanfaatkan secara periodik atau berulang-ulang.

Contoh : layanan konversi mata uang, info cuaca, penterjemahan bahasa dll, seperti contoh gambar dibawah ini



- **Connect existing software**

web service dapat dimanfaatkan untuk mengintegrasikan antar berbagai aplikasi yang berbeda. **Contoh:** aplikasi traveling diintegrasikan dengan aplikasi pemesanan hotel, wisata, rental kendaraan dll meskipun dengan platform, lokasi dan pemrograman yang berbeda. seperti contoh gambar dibawah ini



TEKNOLOGI PENDUKUNG WEB SERVICE

Dalam pengoperasiannya, Web Service menerapkan empat komponen / teknologi yang mendukung kinerjanya. 5 (Lima) komponen tersebut adalah :

- a) XML (*Extensible Markup Language*) merupakan standar untuk markup dokumen yang disahkan oleh W3C (World Wide Web Consortium). XML mendefinisikan sintaks yang umum digunakan untuk markup data secara sederhana. XML menyediakan format standar untuk dokumen komputer. Format ini cukup fleksibel untuk disesuaikan dengan domain yang beragam seperti website, pertukaran data elektronik, grafik vektor, hierarki, serialisasi objek, Remote Procedure Call (RPC), dan sistem voice mail (Harold dan Means, 2002). XML merupakan bagian penting bagi pengembang yang ingin membangun web service. XML berbasis teks, sehingga bersifat platform independent. Sifat ini membuat pertukaran data antar platform menjadi mungkin.

Contoh Dokumen XML :

```
<?xml version="1.0" encoding="utf-8" ?>
<karyawan instansi="UBSI">
  <nama>Rachmat Hidayat, M.Kom</nama>
  <nik> </nik>
  <jabatan>Staf Akademi</jabatan>
  <status>Tetap</status>
  <alamat>Tangerang</alamat>
</karyawan>
```

- b) SOAP (*Simple Object Access Protocol*) adalah sebuah protokol berbasis XML dengan komunikasi bergaya RPC (*Remote Procedure Call*) yang menyediakan cara standar untuk memaketkan pesan (Chappell dan Jewell, 2002). SOAP mendefinisikan cara dalam pembungkusan (encapsulation) dan pertukaran (exchange) pesan.

Dokumen XML yang terdiri dari komponen:

1. **SOAP Envelope**, mengidentifikasi dokumen XML sebagai SOAP message
2. **Elemen Header** (opsional)
3. **Elemen Body**, berisi informasi pemanggilan dan responsnya

4. **Elemen Fault** (opsional), berisi error yang terjadi saat pemanggilan diproses

Contoh Dokumen SOAP Message Dokumen XML yang terdiri dari komponen:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap=" envelope"
                soap:encodingStyle=" encoding">
  <soap:Header> </soap:Header>
  <soap:Body>
    <soap:Fault>.. </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

- c) *Web Service Description Language* (WSDL) adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah Web Service. Di dalam WSDL dijelaskan method-method yang tersedia dalam web service, parameter apa saja yang diperlukan untuk memanggil sebuah method, dan hasil atau tipe data yang dikembalikan oleh method yang dipanggil.

FORMAT WSDL

Elemen <portType>

Mendefinisikan nama web service, operasi- operasi yang dilakukan, dan message yang terlibat

Elemen <message>

Mendefinisikan message yang digunakan oleh web service

Elemen <binding>

Mendefinisikan protokol komunikasi yang digunakan

Elemen <type>

Mendefinisikan tipe data yang digunakan.

```
<types> ... </types>
<message> ... </message>
<portType> ... </portType>
<binding> ... </binding>
<definitions> ... </definitions>
```

- d) *Service Publication and Discovery (UDDI)*, *Universal Description, Discovery and Integration (UDDI)* adalah suatu directory service yang digunakan untuk meregistrasikan dan mencari Web Service.
- e) *Common Internet Protocols (HTTP, TCP/IP)*. Dalam hal ini, HTTP maupun TCP/IP berperan sebagai common internet protocol yang berfungsi sebagai transport layer.

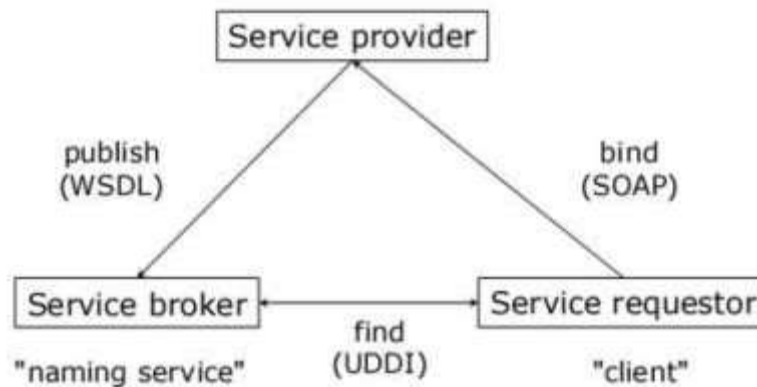
ARSITEKTUR WEB SERVICE



Arsitektur mendeskripsikan struktur: Menurut (Bassetal.1998), Arsitektur dari sistem perangkat lunak (software) terdiri dari strukturnya, dekomposisinya ke dalam komponen-komponen, dan *interface* dan hubungannya. Arsitektur **menjelaskan aspek statis** dan **dinamis** dari sistem software sehingga dapat diperkirakan rancangan bangunan (*building design*) dan diagram alir (*flow chart*) bagi suatu produk software.

Web service memiliki 3(tiga) entitas dalam arsitekturnya, yaitu:

1. **Service Requester** (peminta layanan)
merupakan aplikasi yang bertindak sebagai klien dari Web Service yang mencari dan memulai interaksi terhadap layanan yang disediakan
2. **Service Provider/Broker** (penyedia layanan)
merupakan pemilik Web Service yang berfungsi menyediakan kumpulan operasi dari Web Service.
3. **Service Registry** (daftar layanan)



merupakan tempat dimana Service provider mempublikasikan layanannya. Pada arsitektur Web Service, Service registry bersifat optional. Teknologi web service memungkinkan kita dapat menghubungkan berbagai jenis software yang memiliki platform dan sistem operasi yang berbeda.

CARA KERJA WEB SERVICE

Sisi Server:	Sisi Client:
✓ Membuat fungsi utama/core function ✓ Membuat service wrapper berupa XML-RPC atau SOAP ✓ Membuat deskripsi service berupa WSDL atau instruksi integrasi XML-RPC (memuat semua method public, argumen dan return valuenya); plus dokumentasi yang human readable ✓ Deploy (rilis) service ✓ Daftarkan service tersebut melalui UDDI agar discoverable	✓ Mencari service melalui UDDI ✓ Mengambil service description file berupa WSDL atau instruksi XML-RPC ✓ Membuat klien XML-RPC atau pesan XML untuk dikirim berdasarkan WSDLnya) ✓ Memanggil remote service

OPERASI PADA WEB SERVICE

Secara umum, web service memiliki 3(tiga) operasi yang terlibat di dalamnya, yaitu:

1. **Publish/Unpublish:** Menerbitkan/menghapus layanan ke dalam atau dari registry.
2. **Find:** Service requestor mencari dan menemukan layanan yang dibutuhkan.

3. **Bind:** Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

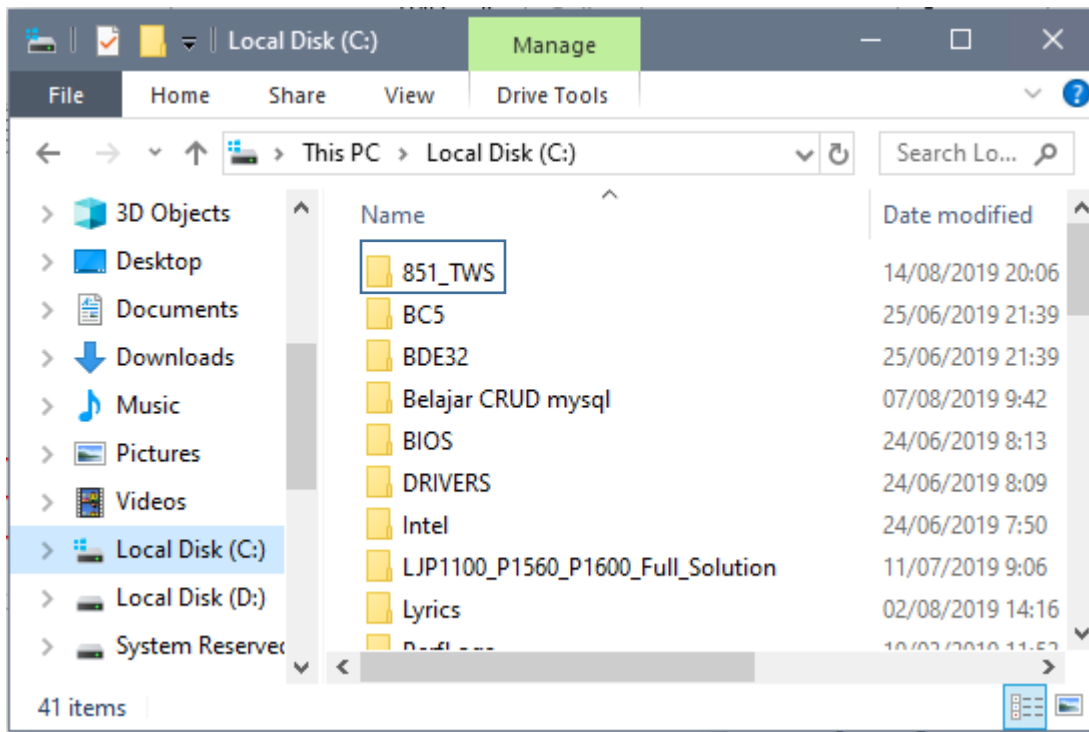
Keuntungan Menggunakan Web Service

- a. *Web Service* menyediakan interoperabilitas antar berbagai aplikasi perangkat lunak yang berjalan pada platform yang berbeda.
- b. *Web Service* menggunakan standard dan protocol yang bersifat terbuka.
- c. *Web service* mengijinkan penggunaan kembali service dan komponen didalam suatu infrastruktur.
- d. *Web service* dapat secara bebas digunakan (loosely coupled) dengan demikian memudahkan suatu pendekatan terdistribusi ke peintegrasian aplikasi

Keunggulan Web Service

- a. Karakteristik standard Web Service saat ini masih dalam tahap perkembangan awal dibandingkan open standard komputer terdistribusi yang lebih matang seperti CORBA. Ini nampaknya akan merupakan suatu kerugian yang temporer ketika kebanyakan vendor sudah merasa terikat dengan standard OASIS untuk menerapkan Mutu dari aspek service dari produk mereka.
- b. Web Service dapat saja memiliki performance/kinerja yang lemah dibandingkan dengan pendekatan komputasi terdistribusi lain seperti RMI, CORBA, atau DCOM. Ini merupakan suatu trade-off yang umum ketika memilih format yang text-based. XML dengan tegas tidak menghitung antar tujuan disain-nya baik singkatan dari penyandian maupun efisiensi dari uraian. Ini bisa berubah dengan standard XML Infoset, yang menguraikan bahasa yang XML-based dalam kaitan dengan hal-hal yang abstrak (unsur-unsur, atribut, logika bersarang). Penyajian angle-bracket (< >) secara tradisional kini dilihat sebagai suatu serialisasi ASCII (atau Unicode) dari XML, bukan XML itu sendiri. Pada model ini, serialisasi biner adalah suatu alternatif yang sama yang sah. Penyajian biner seperti SOAP MTOM menjanjikan untuk meningkatkan efisiensi wire dari XML messaging.

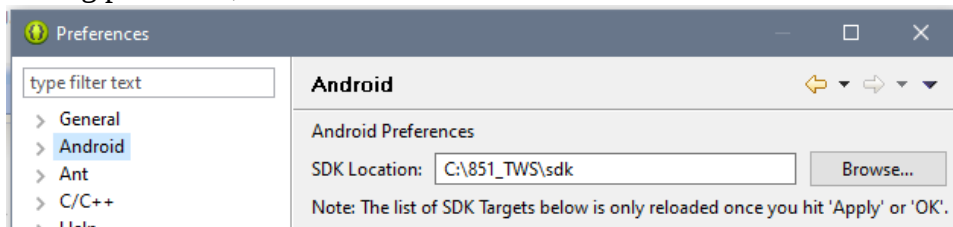
USER INTERFACE (UI)



Install JDK sesuai dengan Operating System Laptop (32bit/64bit)

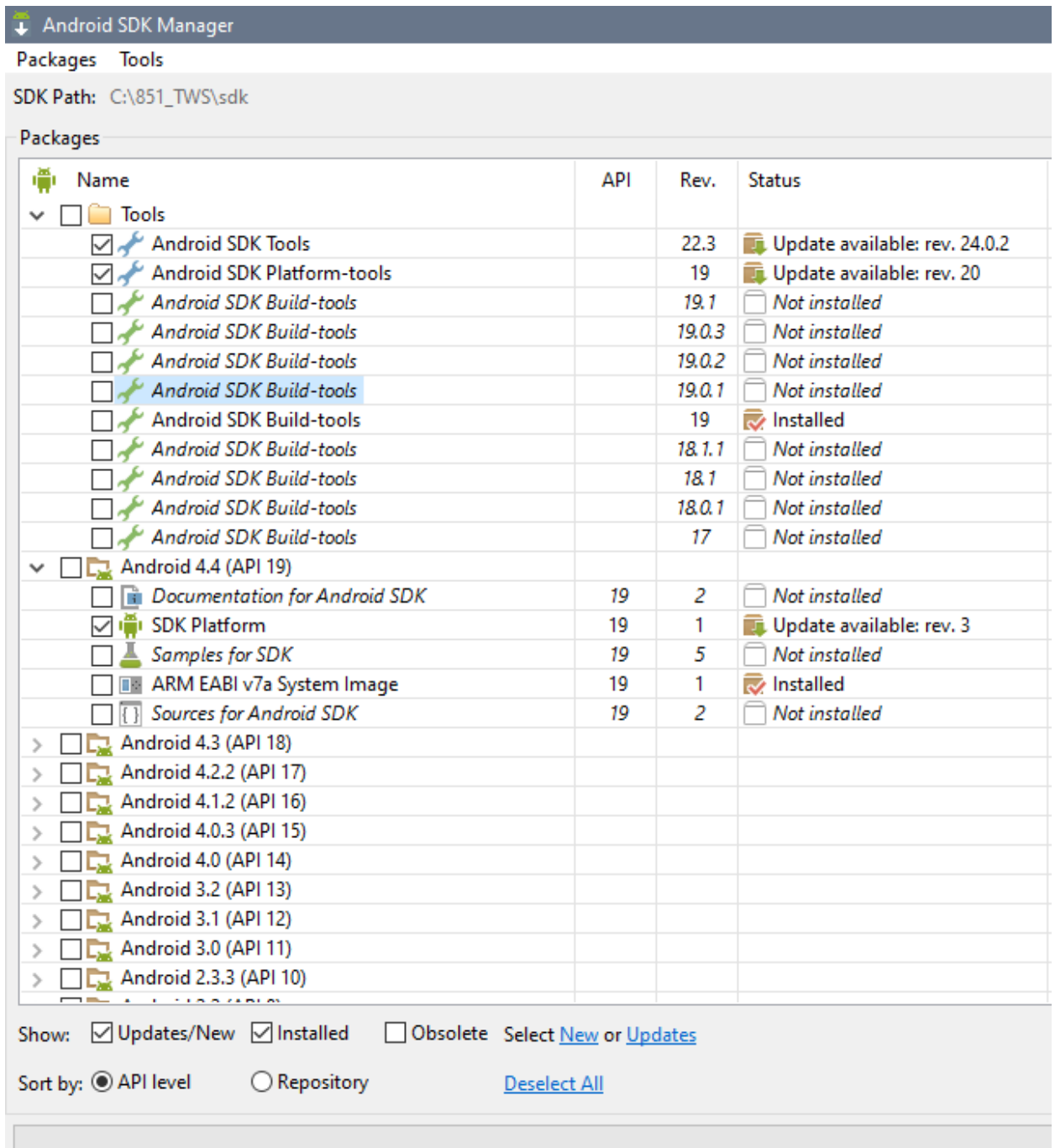
Note: Cek terlebih dahulu apakah sudah terinstall JDK atau belum melalui direktori c:\Program file\Java Setting SDK pada Eclipse

1. Jalankan **Eclipsi** yang berada di folder 851_TWS/,
2. Setting path SDK, melalui menu Window – Preference



Note: pastikan SDK Location di C:\851_TWS\sdk

3. Cek SDK Manager melalui menu Window – Android SDK Manager

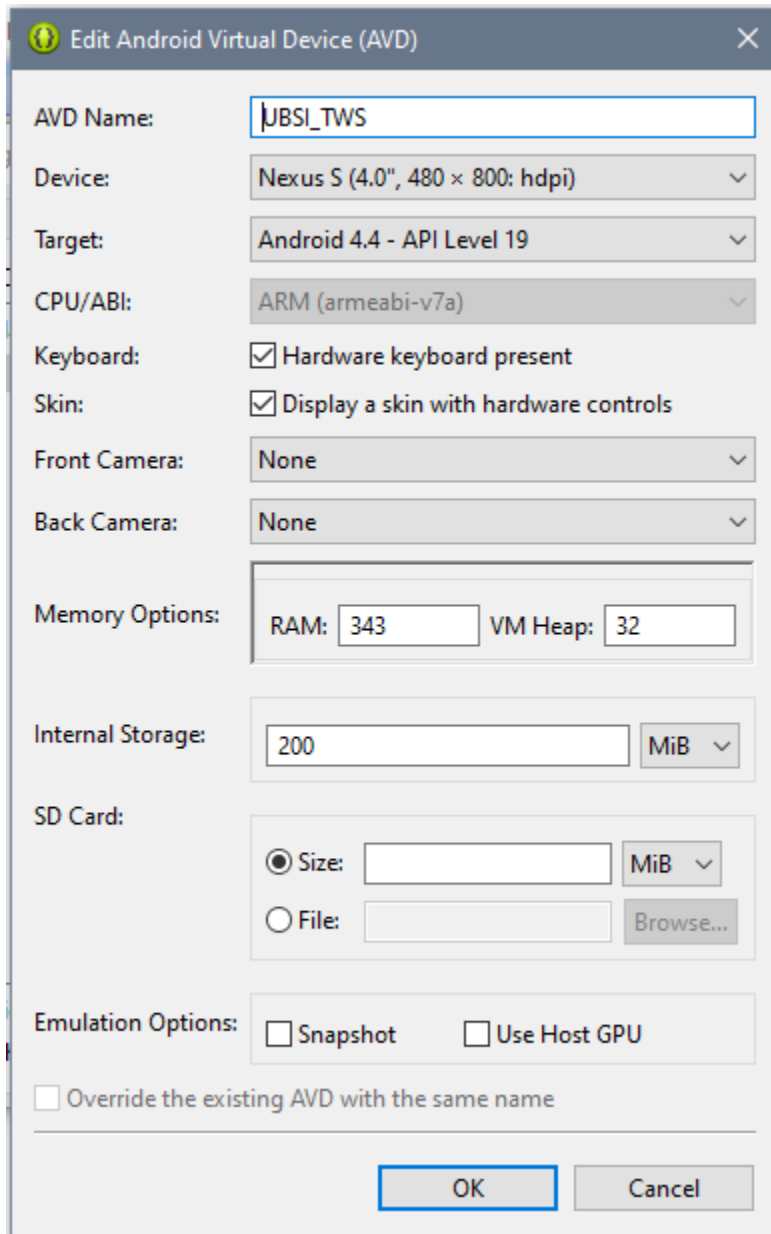


Note: pastikan yang terchecklist pada gambar diatas sudah terinstall pada laptop

4. Buat Android Virtual Device (AVD) Manager sebagai emulator dari program.

Window - Android Virtual Device Manager. Pada tab Android Virtual Device pilih New.

Atur sesuai dengan gambar:



A. Komponen UI

Arsitektur User Interface (UI) pada aplikasi Android adalah User Interface yang meliputi Activity dan User Interface yang terdiri dari komponen. Semua yang berhubungan dengan user interface pada android biasanya berada pada lokasi `res/layout/filename.xml` dimana coding java untuk memanggilnya yang dikenal dengan `R.layout.filename`.

Secara umum sintaks file ini mempunyai struktur sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<ViewGroup
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+[package:]id/resource_name"
    android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
    android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
    [ViewGroup-specific attributes] >
    <View
        android:id="@+[package:]id/resource_name"
        android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
        android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
        [View-specific attributes] >
        <requestFocus/>
    </View>
</ViewGroup>
<View/>
</ViewGroup>
<include layout="@layout/layout_resource" />
</ViewGroup>
```

Komponen-komponennya sebagai berikut:

<ViewGroup>

Kumpulan view dimana dapat menentukan tata letak komponen view secara berbeda, seperti *LinierLayout*, *RelativeLayout*, *FrameLayout* serta *Tabulasi*. Atribut dari ViewGroup ini biasanya terdiri dari:

- **android:id**
resource id, yang berisi variabel unik dari element tersebut.
- **android:layout_height**
dimensi valuenya (height) yang diikutinya dengan opsi “fill_parent” atau “wrap_content”.
- **android:layout_width**
dimensi valuenya (width) yang diikutinya dengan opsi “fill_parent” atau “wrap_content”.

<View>

Sama seperti <ViewGroup> tetapi <View> lebih dikenal dengan “individual UI component”, atributnya juga minimal terdiri dari tiga yang sama persis dengan atribut yang dimiliki oleh <ViewGroup>.

<requestFocus>

Element kosong yang bisa didefinisikan di dalam <view>

<include>

Measukkan file layout ke dalam layout. Atributnya sama dengan <ViewGroup> dan <View> tetapi ada satu tambahan atribut yaitu <resource> yang berfungsi untuk menentukan file layoutnya.

AndroidManifest.xml

File AndroidManifest.xml diperlukan oleh setiap aplikasi android, file ini berada pada folder *root* aplikasi. File ini mendeskripsikan variabel global dari paket aplikasi yang digunakan, file ini juga berfungsi untuk mendeskripsikan resource apa saja yang akan digunakan oleh project seperti koneksi internet, sms, mengakses gps, dll. Berikut adalah contoh file AndroidManifest.xml yang akan kita gunakan dalam program UICoding.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.wilis.uicoding1"
android:versionCode="1"
android:versionName="1.0">
<application android:icon="@drawable/icon"
android:label="@string/app_name">
<activity android:name=".uicoding1"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-sdk android:minSdkVersion="8" />
</manifest>
```

Di dalam AndroidManifest.xml inilah berisi semua pendefinisian hal-hal yang dibutuhkan oleh aplikasi android kita, beberapa elemen yang terdapat di dalam file ini sebagai berikut:

<manifest>

Titik root utama dari AndroidManifest.xml , berisi atribut package aplikasi serta paket *activity* dalam program android kita.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.wilis.uicoding1"
android:versionCode="1"
android:versionName="1.0">
```

Disini terlihat bahwa package kita adalah com.wilis.uicoding1 beserta version code yang digunakan.

<uses-permission>

Menjelaskan tentang user permission/security permission yang harus diberikan agar aplikasi dapat berjalan sebagaimana mestinya misalnya, apabila kita menggunakan resources yang tersedia dari sistem, seperti ketika mengirim sms dan sebagainya. Contoh uses-permission ini adalah

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

<permission>

Menjelaskan batasan tentang user permission/security permission.

<instrumentation>

Mendeklarasikan komponen instrumen yang tersedia untuk menguji fungsionalitas dari paket aplikasi yang digunakan dalam aplikasi android.

<application>

Elemen root yang berisi deklarasi aplikasi android kita.

```
<application android:icon="@drawable/icon">
```

<intent filter>

Mendeklarasikan intent yang dibutuhkan oleh aplikasi android yang digunakan, atribut-atribut bisa diberikan disini untuk supply label, icon, data dan informasi yang kita gunakan dalam aplikasi android.

<action>

Berisi tentang action type yang didukung oleh komponen-komponen yang berada dalam aplikasi android.

<category>

Mendeklarasikan kategori-kategori yang didukung oleh aplikasi android.

<data>

Mendeklarasikan tipe MIME, URL, authority penggunaan URL serta penentuan path yang digunakan dalam URL.

<meta-data>

Mendeklarasikan meta data yang kita butuhkan sebagai tambahan data yang ada kita gunakan dalam aplikasi android.

<receiver>

Mendeklarasikan dimana aplikasi kita diberikan informasi mengenai sesuatu perubahan atau aksi yang terjadi, seperti menerima SMS.

<service>

Mendeklarasikan komponen yang dapat berjalan sebagai service (berjalan di background)

<provider>

Mendeklarasikan komponen-komponen yang mengelola data dan mempublikasikannya untuk dikelola/dipakai oleh aplikasi lain.

<uses-sdk uses-sdk android:minSdkVersion = ?>

Mendeklarasikan sdk android yang digunakan, bisa juga menentukan sdk minimum yang digunakan. Beberapa nilai minSdkVersion sebagai berikut:

- 1 mewakili Android 1.0 SDK
- 2 mewakili Android 1.1 SDK
- 3 mewakili Android 1.5 SDK
- 4 mewakili Android 1.6 SDK
- 5 mewakili Android 2.0 SDK
- 6 mewakili Android 2.0 SDK
- 7 mewakili Android 2.0 Update (2.0.1) SDK
- 8 mewakili Android 2.2 SDK
- 9 mewakili Android 4.2.2 SDK
- 10 mewakili Android 4.2.2.3 SDK
- 11 mewakili Android 3.0 SDK
- 12 mewakili Android 3.1 SDK
- 13 mewakili Android 3.2 SDK
- 14 mewakili Android 4.0 SDK
- 15 mewakili Android 4.0.1 SDK
- 16 mewakili Android 4.1 SDK
- 17 mewakili Android 4.2.1 SDK

B. Pengenalan UI

1. Textview dan Button

Buatlah project android dengan ketentuan sebagai

- | | |
|-----------------------|------------------------------------|
| berikut: Project Name | : Uicoding1 |
| Target SDK | : API 17: Android 4.2 (Jelly Bean) |
| Application Name | : Uicoding1 |
| Package name | : com.wilis.uicoding1 |
| Activity Name | : Uicoding1 |
| Min Required SDK | : API 8: Android 2.2 (Froyo) |

Masuk ke bagian **src** , **Uicoding1.java** :

```
package com.wilis.uicoding1;

import android.os.Bundle;
import android.app.Activity;

public class Uicoding1 extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_uicoding1);
    }

}
```

Uicoding1.java di atas berarti hanya menampilkan isi dari activity_uicoding1.xml, hal itu ditunjukkan dengan coding `setContentView(R.layout.activity_uicoding1)`, itu berarti user interface aplikasi di- handle sepenuhnya di activity_uicoding1.xml.

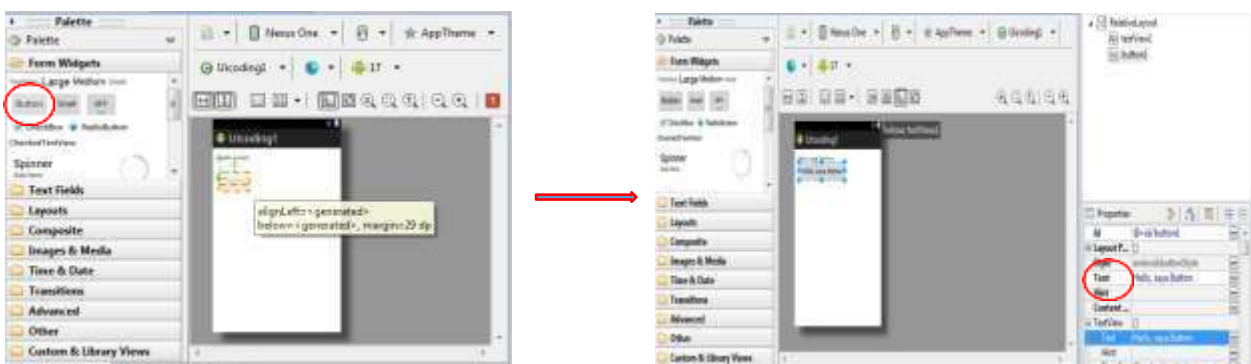
activity_uicoding1.xml

Halaman ini terdapat di bagian **res/layout** . Di dalam activity_uicoding1.xml ini terdiri dari dua Komponen yaitu TextView dan Button. TextView berupa tulisan “ Hello, saya TextView” dengan nama variable id/text serta Button dengan tulisan “Hello, saya Button” dengan variabel id/button. Kedua komponen tersebut ditampilkan dengan layout “wrap_content” yang akan membuat layout untuk elemen tersebut selebar dan setinggi tulisan dari element tersebut. Sementara fill_parent akan membuat tulisan mengisi layar secara penuh.

Untuk pemberian nama variabel dari setiap komponen didalam activity_uicoding1.xml diawali dengan tanda **@+** .

Cara menambahkan Button atau text melalui tampilan **Graphical Layout**:

Klik pada bagian Form Widget dan Drag pada bagian tampilan. Lalu ubah pada bagian properties TextView [Text]. “Hello, saya TextView” dan “Hello, Saya Button”



```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".Uicoding1" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, saya TextView" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:text="Hello, saya Button" />

</RelativeLayout>

```

strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Uicoding1</string>

</resources>

```

Pada bagian **res/values**, strings.xml berisi satu variabel yaitu variabel `app_name`. Variabel ini dipanggil oleh coding di file `AndroidManifest.xml` dengan perintah `android:label="@string/app_name"` untuk menampilkan tulisan `uicoding1` sebagai tampilan nama aplikasi kita. Atau bisa juga menghapus file `strings.xml` dengan mengganti langsung pada file `AndroidManifest.xml` seperti `android:label="uicoding1"` sehingga file `AndroidManifest` tidak membutuhkan `strings.xml`.

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wilis.uicoding1"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.wilis.uicoding1.Uicoding1"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

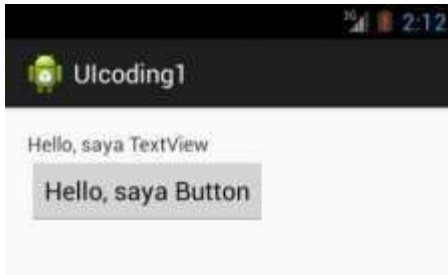
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

package="com.wilis.uicoding1" adalah definisi package dari aplikasi Uicoding1 kita.

Apabila Aplikasi Uicoding1 dijalankan, maka klik kanan pada Uicoding1-> Run As -> Android Application.



2. TextBox, Radio Button dan Check Box

Buatlah project android dengan ketentuan sebagai

berikut: Project Name	: Biodata
Target SDK	: API 17: Android 4.2 (Jelly Bean)
Application	: Biodata
Name Package	: com.wilis.biodata
name Activity	: Biodata
Name Min	: API 8: Android 2.2
Required SDK	(Froyo)

Untuk target SDK dan Min Required SDK Menyesuaikan setup eclips yang dipergunakan

Buka activity_biodata.xml dibagian **res/layout**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".Biodata" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

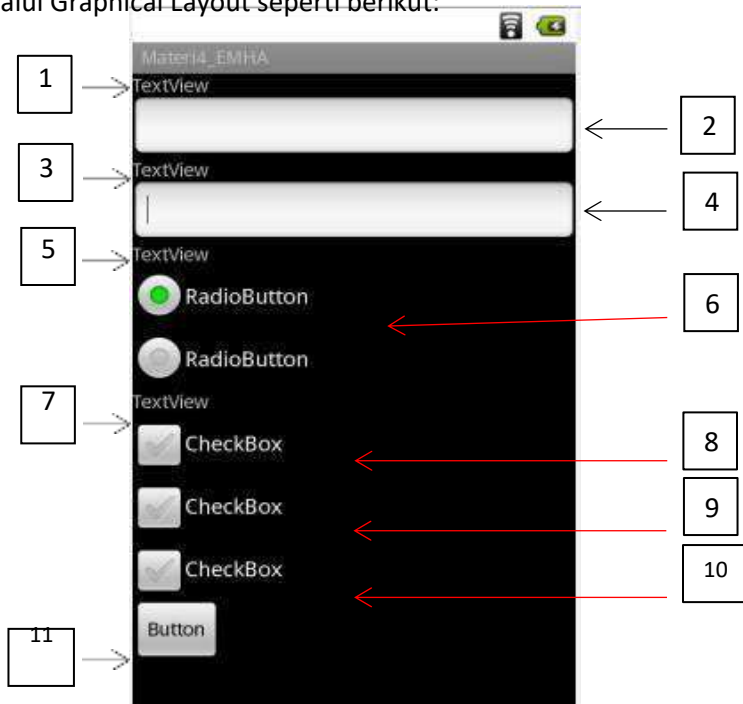
</RelativeLayout>
```

Hapus bagian <TextView>, sehingga menjadi :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".Biodata" >

</RelativeLayout>
```

Desain tampilan melalui Graphical Layout seperti berikut:



Lakukan pengaturan properties sebagai berikut:

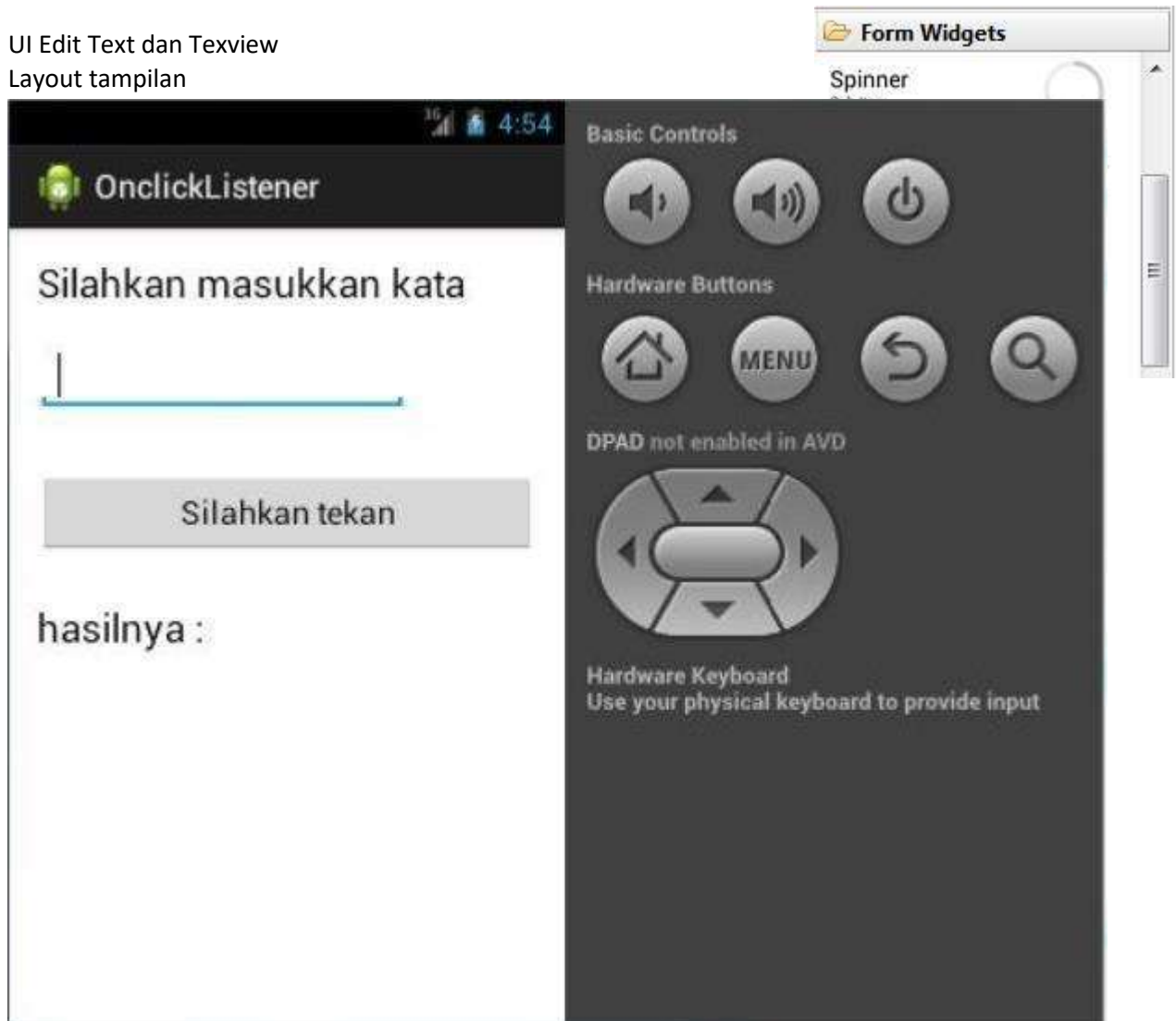
No	Type	Properties	Nilai
1	TextView	Id	@+id/textNama
		Text	Nama Lengkap
2	Edit Text(Plain Text)	Id	@+id/editNama
3	TextView	Id	@+id/textEmail
		Text	Email
4	Edit Text(Plain Text)	Id	@+id/editEmail
5	TextView	Id	@+id/textJenisKelamin
		Text	Jenis Kelamin
6	Radio Group		
	a. Radio Button	Id	radioLaki
		Text	Laki-laki
	a. Radio Button	Id	radioPerempuan
		Text	Perempuan
7	Text View	Id	@+id/textHobi
		Text	Hobi
8	CheckBox	Id	@+id/checkbaca
		Text	Membaca
9	CheckBox	Id	@+id/checkmemancing
		Text	Memancig
10	CheckBox	Id	@+id/checkbersepeda
		Text	Bersepeda
11	Button	Id	@+id/btnok

		Text	OK
--	--	------	----

Untuk RadioGroup:

D. UI Edit Text dan Textview

1. Layout tampilan



2. Berdasarkan contoh yang mahasiswa sudah praktikan pada materi sebelumnya, mahasiswa harus sudah mampu secara mandiri mempersiapkan proyek baru:

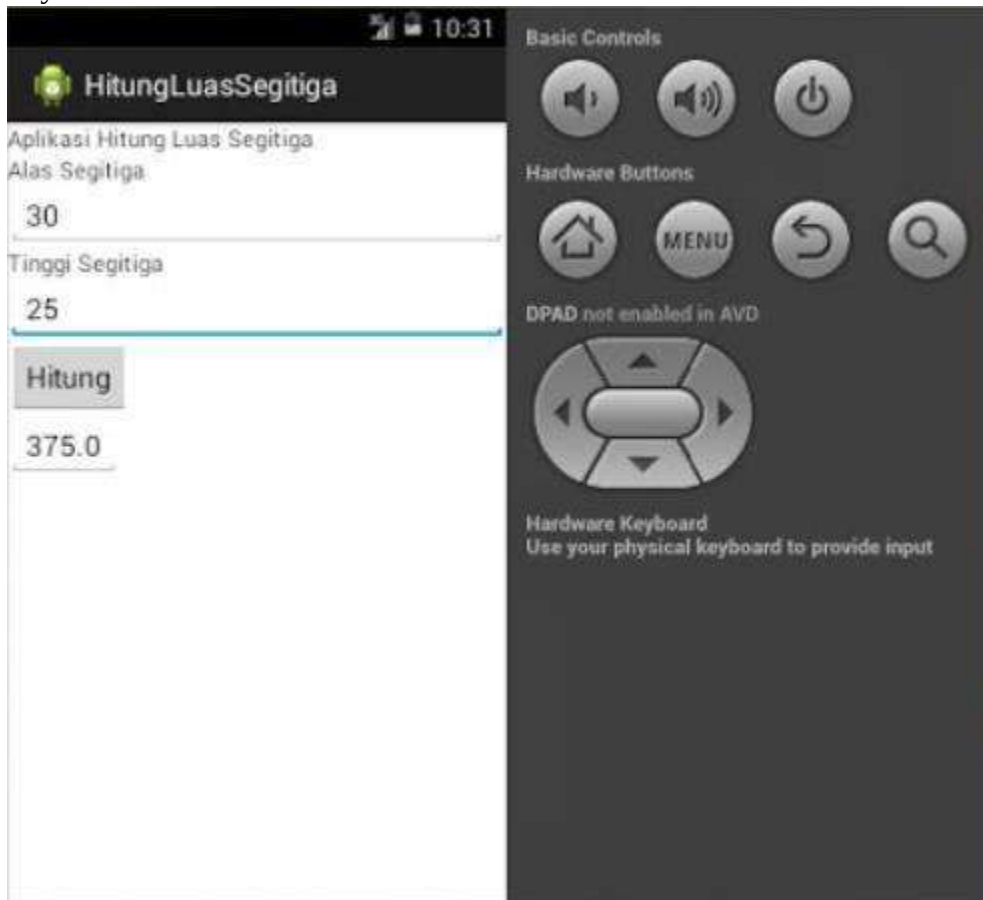
- a. Project Name :
- b. Target SDK :
- c. Application Name :
- d. Package name :
- e. Activity Name :
- f. Min Required SDK :

Pahami dulu script pada point 4, agar mahasiswa dapat menentukan point 2a-2f

3. Buat UI seperti point 1
4. Coding
Script untuk menghasilkan aplikasi pada point 1 melalui laman : <http://fahrur-blogku.blogspot.com/2014/07/membuat-edittext-dan-textview-pada.html>
5. Uji coba aplikasi
6. Mahasiswa wajib mempersiapkan penguasaan materi dari script tersebut yang akan dipergunakan dalam diskusi pada menu e-learning

E. UI Matematika

1. Layout



2. Berdasarkan contoh yang mahasiswa sudah praktikan pada materi sebelumnya, mahasiswa harus sudah mampu secara mandiri mempersiapkan projek baru:
 - a. Project Name :
 - b. Target SDK :
 - c. Application Name :
 - d. Package name :

- e. Activity Name :
- f. Min Required SDK :
3. Source Code I. Lengkapi script bawaan pada saat pembuatan proyek, sehingga menghasilkan script lengkap seperti dibawah ini

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Aplikasi Hitung Luas Segitiga"
    />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Alas Segitiga"/>
    <EditText android:id="@+id/eAlas"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
    />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Tinggi Segitiga"/>
    <EditText android:id="@+id/eTinggi"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
    />
    <Button android:id="@+id/bSegitiga"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hitung"/>
    <EditText
        android:id="@+id/eHasil"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:editable="false"/>
</LinearLayout>
```

4. Source Code II. Lengkapi script bawaan pada saat pembuatan proyek, sehingga menghasilkan script lengkap seperti dibawah

```
package com.amrid.hitungluassegitiga;

import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class ActivityUtamaLuas extends Activity implements OnClickListener {

    EditText eAlas;

    EditText eTinggi;

    EditText eHasil;

    Button bSegitiga;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_utama_luas);

        eAlas = (EditText)findViewById(R.id.eAlas);
        eTinggi = (EditText)findViewById(R.id.eTinggi);
        eHasil = (EditText)findViewById(R.id.eHasil);
        bSegitiga = (Button)findViewById(R.id.bSegitiga);

        bSegitiga.setOnClickListener(this);
    }
}
```

```
@Override
public void onClick(View view) {
    // TODO Auto-generated method stub
    double Alas = Double.parseDouble(String.valueOf(eAlas.getText()));
    double Tinggi = Double.parseDouble(String.valueOf(eTinggi.getText()));
    double Hasil = 0.5*Alas*Tinggi;

    eHasil.setText(String.valueOf(Hasil));

}

}
```

5. Uji coba aplikasi
6. Mahasiswa wajib mempersiapkan penguasaan materi dari script tersebut yang akan dipergunakan dalam diskusi pada menu e-learning

RESTFUL WEB SERVICE

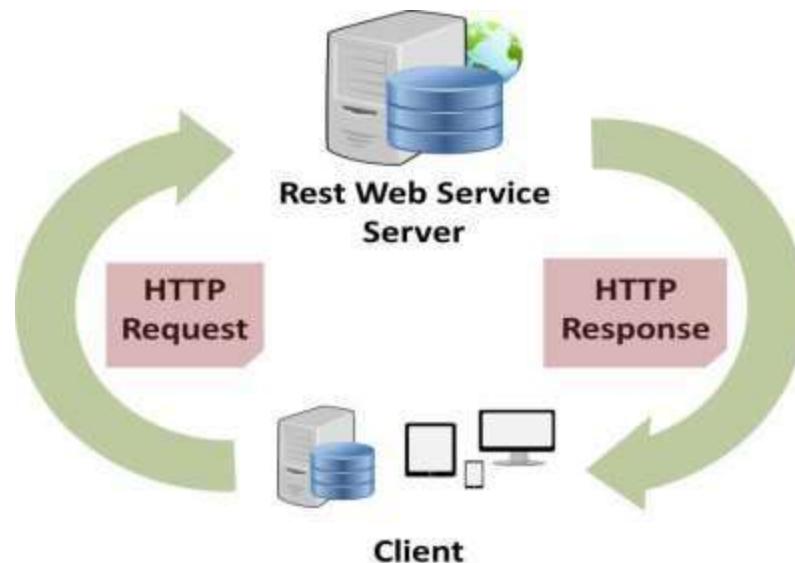
A. Pengenalan Restful Web Service

Salah satu web service yang biasa digunakan yaitu REST atau biasa disebut RESTful Web Service. REST sendiri merupakan singkatan dari *Representational State Transfer*, yaitu suatu gaya arsitektur perangkat lunak dalam menyediakan sumber daya /data (*resources*) pada REST server, serta dapat diakses dan ditampilkan *resources* tersebut untuk dapat digunakan pada REST Client.

Layanan Restful Web Service menggunakan metode HTTP dalam menerapkan konsep arsitektur REST, dimana setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, yang pada umumnya menggunakan JSON atau XML. Berikut beberapa metode HTTP yang umum digunakan dalam arsitektur Restful Web Service.

- **GET**, hanya menyediakan akses pembacaan pada *resource*
- **PUT**, dapat digunakan untuk memperbaharui *resource* baru
- **DELETE**, dapat digunakan untuk menghapus *resource*
- **POST**, dapat digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru

Berikut gambaran ini cara kerja Restful Web Service



Sebuah Client mengirimkan atau meminta sebuah data melalui HTTP Request, lalu kemudian server merespon melalui HTTP Response. Komponen dari HTTP Request bisa terdiri dari HTTP Method (GET, POST, DELETE, PUT, dan yang lainnya), *Uniform Resource Identifier* (URI) untuk mengidentifikasi lokasi *resource* pada server. Sementara komponen HTTP Response bisa terdiri dari *Status/Response Code* yang mengindikasikan status server terhadap *resource* yang direquest (misal : 404, artinya *resource* tidak ditemukan dan 200 response OK)

B. Kebutuhan Tools

Dalam implementasi RESTful Web Service pada modul pembelajaran ini dibutuhkan beberapa hal yang harus dipersiapkan pada PC/laptop, diantaranya yaitu:

1. Xampp sebagai web server, Sublime/atom sebagai editor (*pastikan dilaptop mahasiswa sudah terinstall tools tersebut, kelanjutan dari semester IV*)
2. Codeigniter dan library REST server yang diperlukan dapat diunduh di <https://github.com/chriskacerguis/codeigniter-restserver> untuk versi terbaru, versi yang digunakan disini adalah <https://github.com/ardisaurus/ci-restserver>.

C. Implementasi Restful Web Service

Dibawah ini akan dijabarkan contoh implementasi Restful Web Service menggunakan Framework Codeigniter. Dalam konsep MVC Codeigniter, *controller* adalah titik pusat logika dimana dipanggil ketika pengguna membuat permintaan dan kemudian berdasarkan logika di *controller* itu mengambil data dan output dapat dilihat atau ditampilkan. Namun, dalam penerapan RESTful diperlukan library REST_Controller sebagai pengaturan logikanya.

1. Konfigurasi Database

- Buat database baru dengan nama **db_member**:

```
CREATE DATABASE db_member;
```

- Dalam database db_member, buat sebuah tabel dengan nama **member**.

Nama <i>Field</i>	Type	Size	Ket
Id	Int	11	Auto_Increment Not Null, Primary Key
nama_member	Varchar	40	Not Null
Email	Varchar	35	Not Null
no_telp	Varchar	16	Not Null

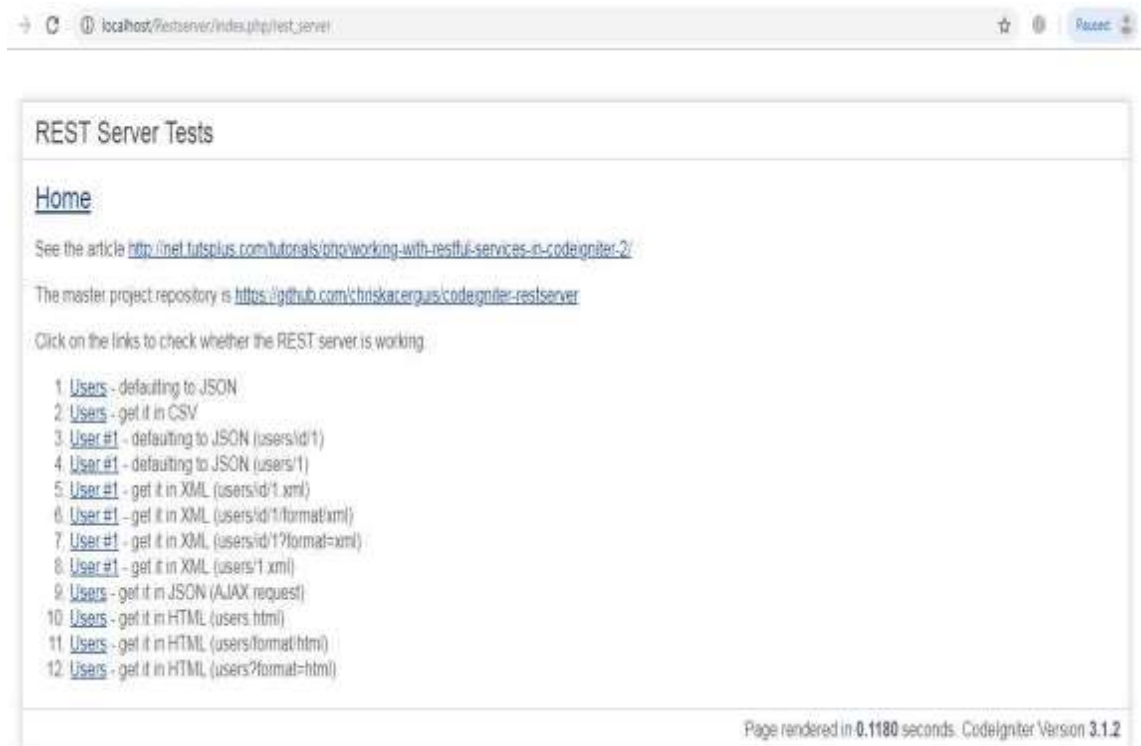
```
CREATE TABLE IF NOT EXISTS `member` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_member` varchar(40) NOT NULL,
  `email` varchar(35) NOT NULL,
  `no_telp` varchar(16) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- Pilih tabel member, lalu masukan beberapa isi *record*.

```
INSERT INTO `member` (`nama_member`, `email`, `no_telp`) VALUES
('Tiwi Pertiwi','tiwi@gmail.com','111122223333'),
('Hanna Putri','hanna@gmail.com','555566667777'),
('Mus Dalifa','mus@gmail.com','888899990000');
```

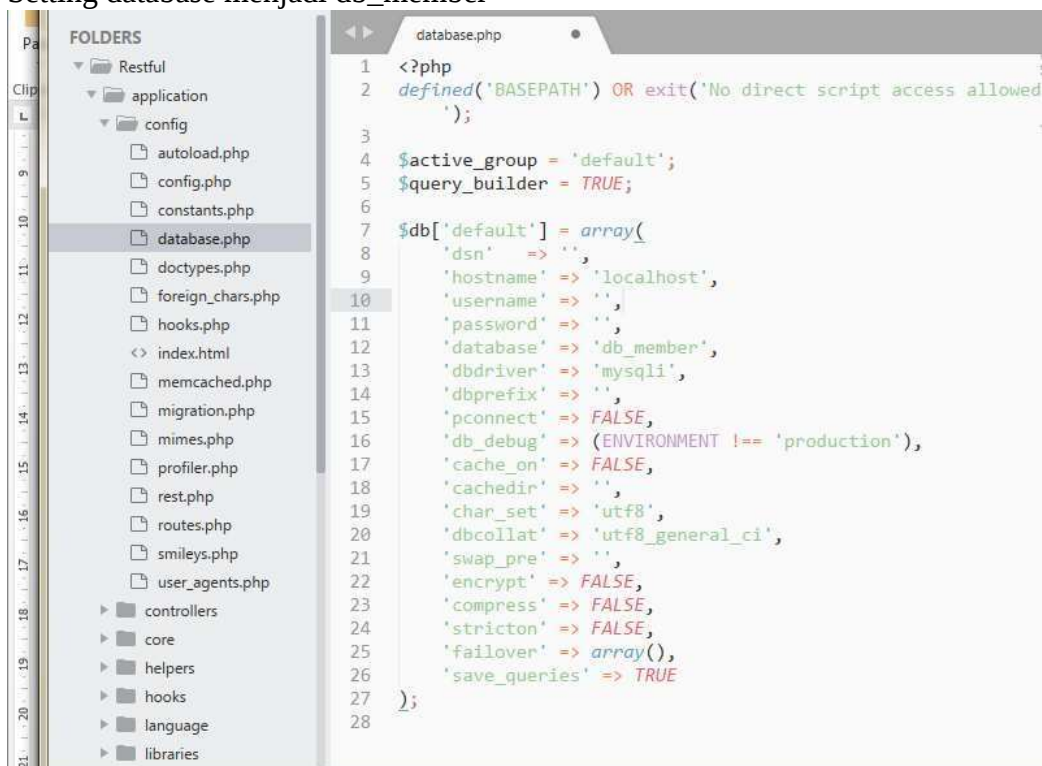
2. Persiapan Project RESTful Web Service

- Pastikan Folder master Codeigniter yang disertai library REST Server telah ada pada folder C:/xampp/htdocs dilaptop masing-masing. Dengan nama folder diubah dan diberi nama **Restserver**.
- Pastikan service Apache dan Mysql pada Xampp Control panel telah aktif.
- Jalankan Browser, lalu ketikan alamat : **localhost/Restserver/index.php/rest_server**
- Hingga muncul tampilan dibawah ini untuk memastikan project Restserver dan library REST Server berhasil running.



3. Setting File Koneksi Database

- Jalankan Editor Sublime Text, lalu buka Folder Project **Restful**.
- Buka File database.php pada folder application/config/database.php
- Setting database menjadi db_member



4. Implementasi Metode GET

Untuk melakukan implementasi Restful Web Service dengan metode GET, POST, PUT, dan DELETE, terlebih dahulu buat controller baru dengan nama **“Rest_member.php”** pada folder controller.

Kemudian, untuk metode GET, ketikkan script dibawah ini:

```
<?php
// Dokumentasi Pengerjaan Kelompok
// Nama kelompok : .....
// Kelas      : .....
// Ketua      : Rudi Sujarwo - 12181234
// Anggota : Joko Susanto - 12189388
//           : ....
//           : dst..

defined('BASEPATH') OR exit('No direct script access allowed');
require APPPATH . '/libraries/REST_Controller.php';
use Restserver\libraries\REST_Controller;

class Rest_member extends REST_Controller {

    function __construct($config = 'rest'){
        parent::__construct($config);
        $this->load->database();
    }

    function index_get() {
        // Deskripsi Fungsi :
        // -----
        // Script dibawah ini merupakan implementasi dari metode GET
        // -----
        $id = $this->get('id');
        if ($id == "") {
            $member = $this->db->get('member')->result();
        } else {
            $this->db->where('id', $id);
            $member = $this->db->get('member')->result();
        }
        $this->response($member, 404);
    }

}
?>
```

Penjelasan Code :

```
require APPPATH . '/libraries/REST_Controller.php';
use Restserver\libraries\REST_Controller;
```

Script diatas menunjukan pemanggilan Controller REST_Controller.php yang ada pada libraries untuk web service. *(setiap pembuatan controller baru, wajib disertakan deklarasi controller library REST web service)*

```
function __construct($config = 'rest'){
    parent::__construct($config);
    $this->load->database();
}
```

Script diatas menunjukan function dengan parameter \$config yang diberi nilai 'rest'. Pemberian nilai rest merujuk pada function construct file REST_Controller(ada di folder libraries), jika pengaturasn REST web service ada, maka file database.php akan dijalankan.

```
// Deskripsi Fungsi :
// .....
// Script dibawah ini merupakan implementasi dari metode GET
// .....
$id = $this->get('id');
if ($id == "") {
    $member = $this->db->get('member')->result();
} else {
    $this->db->where('id', $id);
    $member = $this->db->get('member')->result();
}

$this->response($member, 404);
```

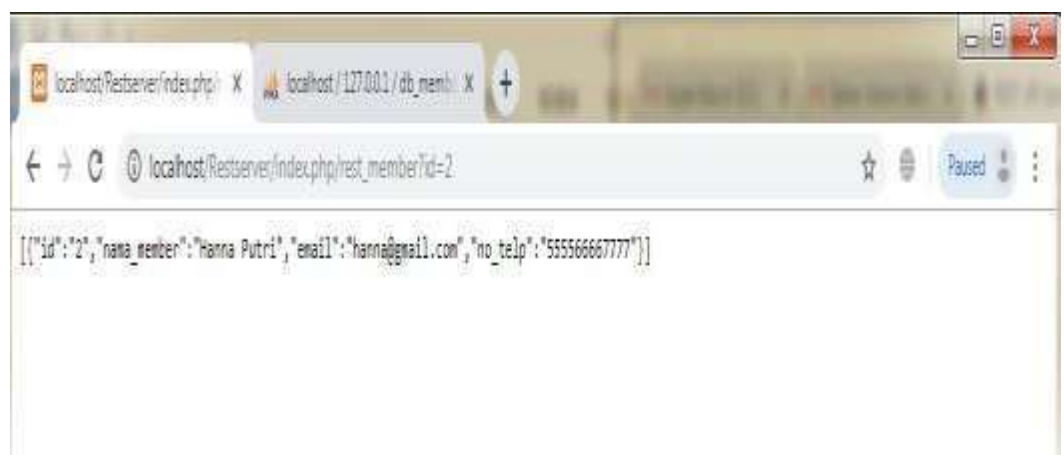
Script diatas menunjukan pembuatan fungsi get, yang akan memeriksa apakah ada property “id” pada address bar, jika tidak ada maka data akan ditampilkan semua tanpa ada seleksi.

Tampilan implementasi pada browser :

- Ketika tidak ada property id pada addrees bar :
localhost/Restserver/index.php/rest_member



- Ketika ada property id pada address bar :
localhost/Restserver/index.php/rest_member?id=2



5. Implementasi Metode POST

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi function index_get.

Kemudian, untuk metode POST, tambahkan script dibawah ini diatas script penutup php :

Penjelasan Code :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
require APPPATH . '/libraries/REST_Controller.php';
use Restserver\libraries\REST_Controller;

class Rest_member extends REST_Controller {

    function __construct($config = 'rest'){
        parent::__construct($config);
        $this->load->database();
    }

    function index_get()
    {
        $id = $this->get('id');
        if ($id == '') {
            $member = $this->db->get('member')->result();
        } else {
            $this->db->where('id', $id);
            $member = $this->db->get('member')->result();
        }
        $this->response($member, 200);
    }

    function index_post() {
        $data = array(
            'id' => $this->post('id'),
            'nama_member' => $this->post('nama_member'),
            'email' => $this->post('email'),
            'no_telp' => $this->post('no_telp'));
        $insert = $this->db->insert('member', $data);
        if ($insert) {
            $this->response($data, 200);
        } else {
            $this->response(array('status' => 'fail', 502));
        }
    }

}
?>
```

```
function index_post() {
    $data = array(
        'id' => $this->post('id'),
        'nama_member' => $this->post('nama_member'),
        'email' => $this->post('email'),
        'no_telp' => $this->post('no_telp'));
    $insert = $this->db->insert('member', $data);
    if ($insert) {
        $this->response($data, 200);
    } else {
        $this->response(array('status' => 'fail', 502));
    }
}
```

Script diatas menunjukkan pembuatan fungsi POST, yang digunakan untuk mengirimkan data baru dari client ke REST server. Pada kasus diatas function index_post pada controller rest_member.php untuk menambahkan (*insert*) data pada tabel member dengan atribut id, nama_member, email dan no_telp.

Untuk pengujian metode POST akan dibahas pada bahasan berikutnya mengenai tools postman.

6. Implementasi Metode PUT

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi beberapa function.

Kemudian, untuk metode PUT, tambahkan script dibawah ini diatas script penutup php :

```
function index_put() {
    $id = $this->put('id');
    $data = array(
        'id' => $this->post('id'),
        'nama_member' => $this->post('nama_member'),
        'email' => $this->post('email'),
        'no_telp' => $this->post('no_telp'));
    $this->db->where('id', $id);
    $update = $this->db->update('member', $data);
    if ($update) {
        $this->response($data, 200);
    } else {
        $this->response(array('status' => 'fail', 502));
    }
}

?>
```

Penjelasan Code :

```
function index_put() {
    $id = $this->put('id');
```

```

$data = array(
    'id'          => $this->post('id'),
    'nama_member' => $this->post('nama_member'),
    'email'       => $this->post('email'),
    'no_telp'     => $this->post('no_telp'));
$this->db->where('id', $id);
$update = $this->db->update('member', $data);
if ($update) {
    $this->response($data, 200);
} else {
    $this->response(array('status' => 'fail', 502));
}
}

```

Script diatas menunjukan pembuatan fungsi PUT, yang digunakan untuk memperbaharui data (*update*) dari client ke REST server. Pada kasus diatas function *index_put* pada controller *rest_member.php* untuk memperbaharui (*update*) data pada atribut *nama_member*, *email* atau *no_telp* dengan *id* yang ditunjuk pada tabel *member*.

Untuk pengujian metode PUT akan dibahas pada bahasan berikutnya mengenai tools *postman*.

7. Implementasi Metode DELETE

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi beberapa function.

Kemudian, untuk metode DELETE, tambahkan script dibawah ini diatas script penutup *php* :

```

function index_delete() {
    $id = $this->delete('id');
    $this->db->where('id', $id);
    $delete = $this->db->delete('member');
    if ($delete) {
        $this->response(array('status'=>'sukses'), 200);
    } else {
        $this->response(array('status' => 'gagal', 502));
    }
}

}
?>

```

Penjelasan Code :

```

function index_delete() {
    $id = $this->delete('id');
    $this->db->where('id', $id);
    $delete = $this->db->delete('member');

```

```
if ($delete) {  
    $this->response(array('status'=>'sukses'), 200);  
} else {  
    $this->response(array('status' => 'gagal', 502));  
}  
}
```

Script diatas menunjukan pembuatan fungsi DELETE, yang digunakan untuk menghapus data (*delete*) dari client ke REST server. Pada kasus diatas function `index_delete` pada controller `rest_member.php` untuk menghapus (*delete*) data pada atribut `nama_member`, `email` atau `no_telp` dengan `id` yang ditunjuk pada tabel `member`.

Untuk pengujian metode DELETE akan dibahas pada bahasan berikutnya mengenai tools postman.

PENGUJIAN RESTFUL WEB SERVICE

Pengenalan dan Instalasi Postman

1. Sejarah Singkat Postman

Postman pertama kali dibuat sebagai projek sampingan yang dikerjakan oleh Abhinav Asthana untuk mengatasi tantangan dalam pengujian API, kemudian postman mulai populer pada tahun 2012, yaitu ketika Abhinav selaku CEO dan co-founder Postman dengan ditemani Ankit Sobti dan Abhijit Kane. mengunggah perojek yang dia kerjakan ke Chrome Web Store artinya aplikasi postman masih dalam bentuk plug-in dari web browser Chrome. Namun akhirnya postman berhasil release dalam bentuk aplikasi yang berdiri sendiri dengan user interface yang mirip dengan tampilan web browser. Saat ini postman sudah digunakan oleh 7 juta pengembang dan 300 perusahaan.

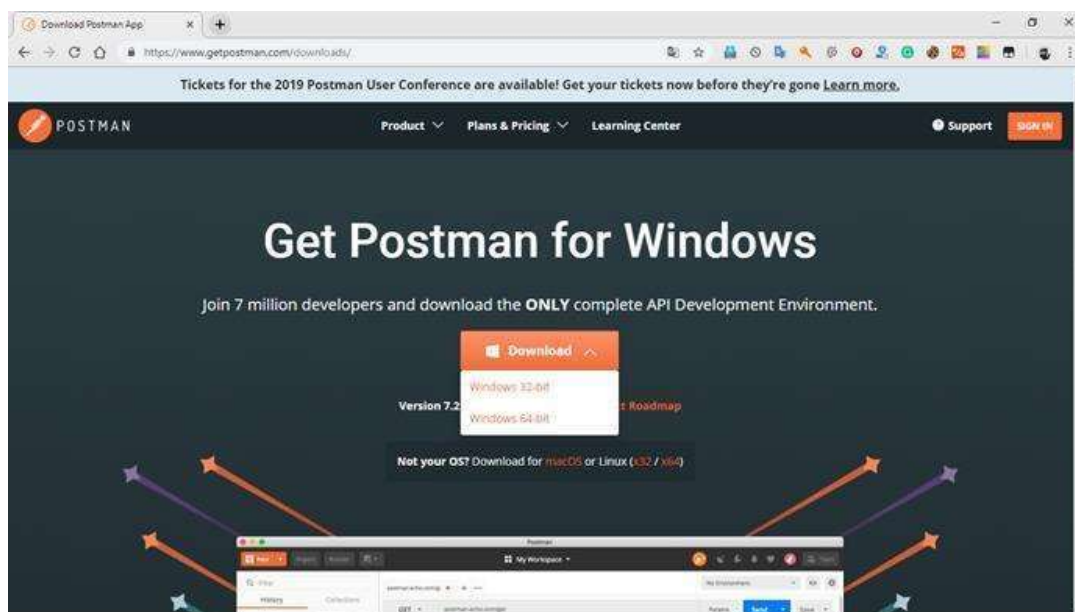
2. Pengertian Postman

Postman adalah sebuah aplikasi fungsinya adalah sebagai REST Client atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat. Postman ini merupakan tools wajib bagi para developer yang bergerak pada pembuatan API, fungsi utama postman ini adalah sebagai GUI API Caller Pemanggil. namun sekarang postman juga

menyediakan fitur lain yaitu Sharing Collection API for Documentation (free), Testing API (free), Realtime Collaboration Team (paid), Monitoring API (paid), Integration (paid) detailnya silahkan dicek disini. Dulu awal pertama kali postman muncul sebagai add on dari Chrome namun sekarang sudah menjadi aplikasi sendiri. Jika kalian sedang membuat API sangat direkomendasikan untuk menggunakan Postman untuk testing API yang kalian buat.

3. Instalasi Postman

Postman tersedia sebagai aplikasi asli untuk sistem operasi macOS, Windows (32-bit dan 64-bit), dan Linux (32-bit dan 64-bit). Untuk mendapatkan aplikasi Postman, dapat diunduh pada website resminya yaitu getpostman.com



atau bisa di download melalui link berikut:

- Linux

https://drive.google.com/file/d/1I-U-7sqAowZhL5dosM3_5B0XS9QMLMMa/view?usp=sharing

- Mac

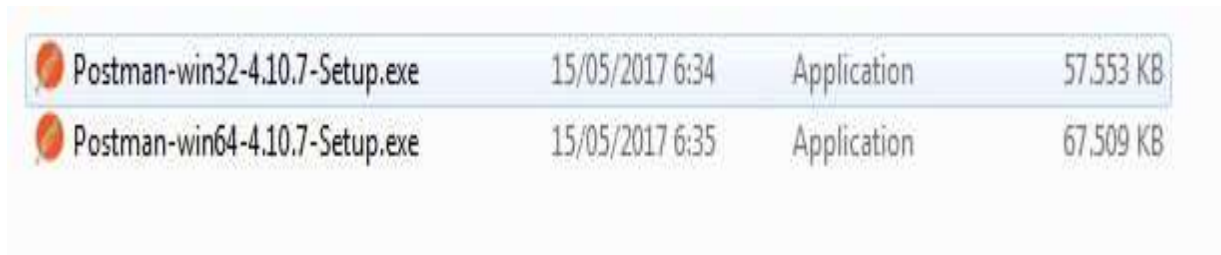
https://drive.google.com/file/d/1nTXd-9cPPXuq4JLx_OBUr9jBWld61bML/view?usp=sharing

- Windows_

[Versi 32 Bit](#)

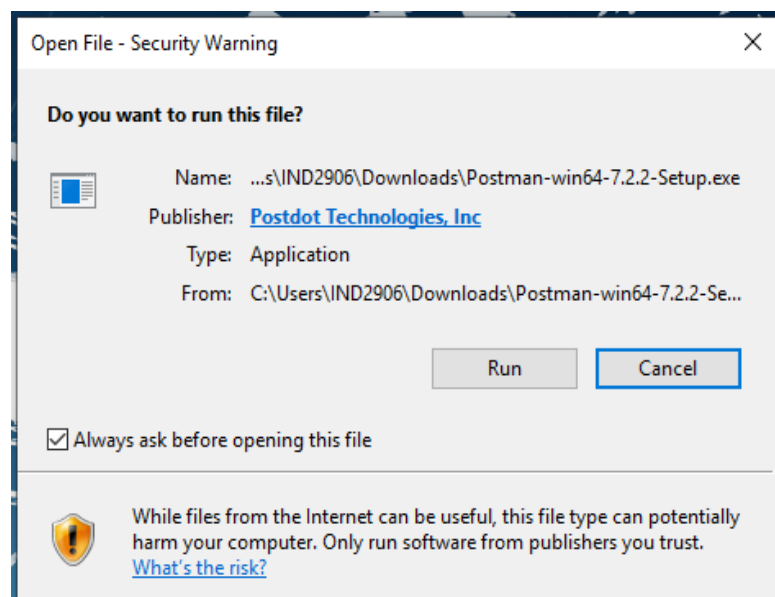
Versi 64 Bit

Setelah postman selesai didownload, selanjutnya jalankan paket instalasi postman dengan cara double klik.

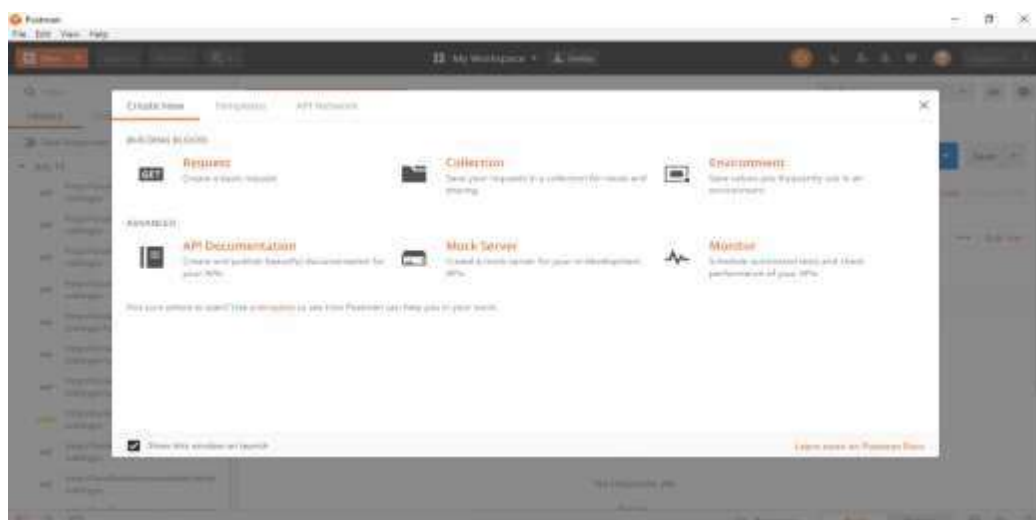


	Postman-win32-4.10.7-Setup.exe	15/05/2017 6:34	Application	57.553 KB
	Postman-win64-4.10.7-Setup.exe	15/05/2017 6:35	Application	67.509 KB

Pilih Run jika muncul popup seperti berikut:



Kemudian tunggu sampai Postman nya terbuka Seperti ini:



Selesai, postman sudah terinstal pada perangkat anda.

B. Implementasi Pengujian menggunakan Postman

Pengujian Restful Web Service menggunakan postman ini perlu ada beberapa persiapan terlebih dahulu. Persiapan yang diperlukan adalah sebagai berikut:

1. Webserver seperti Xampp, Wampp, atau lainnya. Jalankan
2. Konfigurasi Database

Database yang akan digunakan adalah database yang sudah dibuat pada pertemuan sebelumnya yaitu menggunakan database **db_member**

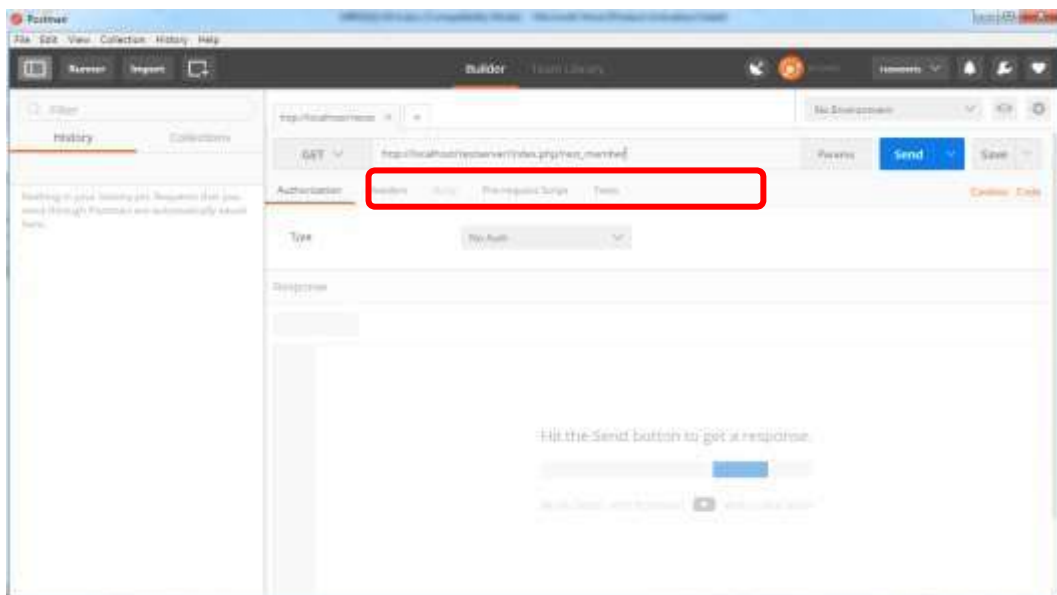
3. Projek program yang ingin diuji

Projek yang akan diuji dalam hal ini adalah projek yang sudah dibuat pada pertemuan minggu sebelumnya juga yaitu tentang restful pendaftaran member.

Setelah persiapan selesai barulah kita akan menguji dengan postman. Langkahnya sebagai berikut:

1. Buka aplikasi postman yang sudah terinstal
2. Paste url pada inputan pada postman. Seperti gambar berikut:

http://localhost/restserver/index.php/rest_member



3. Selanjutnya pilih jenis pengujian datanya. Dalam hal ini yang akan diuji adalah GET, POST, PUT, dan DELETE.

a. Pengujian GET

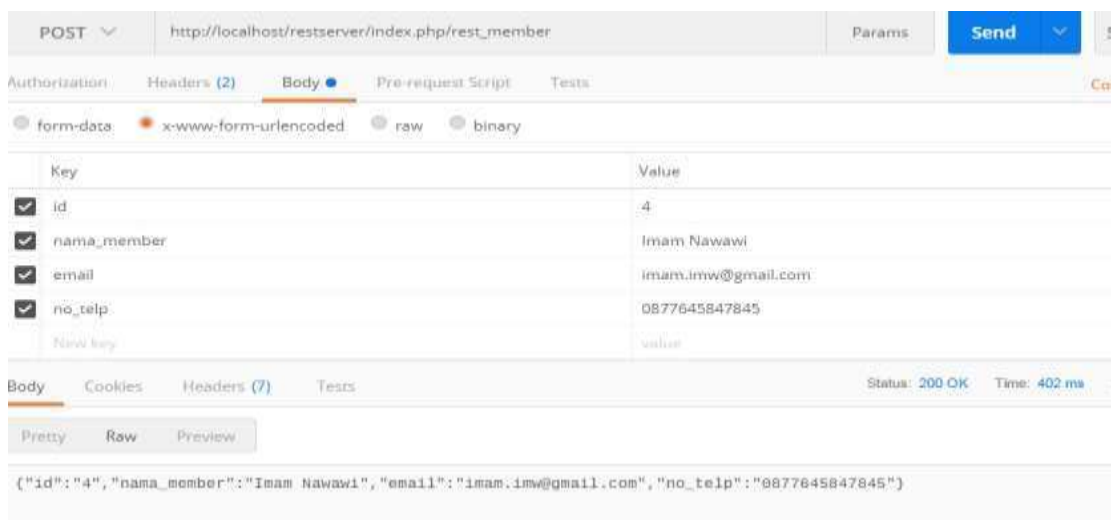
1. pilih GET terlebih dahulu, lalu klik tombol Send
2. Hasilnya seperti berikut:

```
Pretty Raw Preview
```

```
[{"id": "1", "nama_member": "Tiwi Pertiwi", "email": "tiwi@gmail.com", "no_telp": "111122223333"}, {"id": "2", "nama_member": "Hanna Putri", "email": "hanna@gmail.com", "no_telp": "555566667777"}, {"id": "3", "nama_member": "Mus Dalifa", "email": "mus@gmail.com", "no_telp": "888899990000"}]
```

b. Pengujian POST

1. Untuk pengujian POST, maka pilih jenis pengujian POST
2. Setelah pengujian dipilih, selanjutnya klik body yang ada dibagian bawah inputan url di atas, kemudian pilih x-www.form-urlencoded.
3. Selanjutnya isi key sesuai dengan field-field yang ada pada tabel yang digunakan, sedangkan value diisi bebas. Lalu klik tombol send
4. Hasilnya pengujian POST seperti berikut:



c. Pengujian PUT

Pengujian PUT langkahnya sama seperti metode POST. Hasilnya seperti berikut:

PUT http://localhost/restserver/index.php/rest_member Params Send

Key	Value
New key	value

Authorization Headers (2) **Body** Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value
☒ nama_member	Nandi
☒ email	imamkhotib0@gmail.com
☒ no_telp	02175346364
☒ id	3
New key	value

Body Cookies Headers (7) Tests Status: 200 OK Time: 393 ms

Pretty Raw Preview

```
{\"id\": \"3\", \"nama_member\": \"Nandi\", \"email\": \"imamkhotib0@gmail.com\", \"no_telp\": \"02175346364\"}
```

d. Pengujian DELETE

Untuk pengujian delete, langkahnya kurang lebih sama dengan POST dan PUT.

DELETE http://localhost/restserver/index.php/rest_member Params Send

Authorization Headers (2) **Body** Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

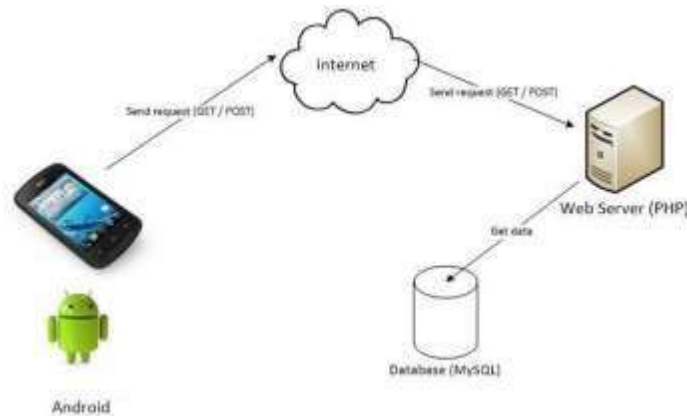
Key	Value
☒ id	1
☒ nama_member	Tiwi Pertiwi
☒ email	tiwi@gmail.com
☒ no_telp	111122223333
New key	value

Body Cookies Headers (7) Tests Status: 200 OK Time: 1604 ms

Pretty Raw Preview

```
{\"status\": \"berhasil\"}
```

IMPLEMENTASI CLIENT SERVER PHP(MYSQL) DAN ANDROID



Sebagai gambaran aplikasi web service yang akan dipraktikan diilustrasikan seperti gambar di atas. Ponsel Android akan melakukan request (get/post) ke server melalui internet. Cara ini juga sama jika ingin aplikasi berada pada localhost. Selanjutnya web server (dalam hal ini PHP), akan memproses request dari Android dan akan melakukan query ke database (MySQL). Untuk menghasilkan konsep di atas dibutuhkan 3 langkah:

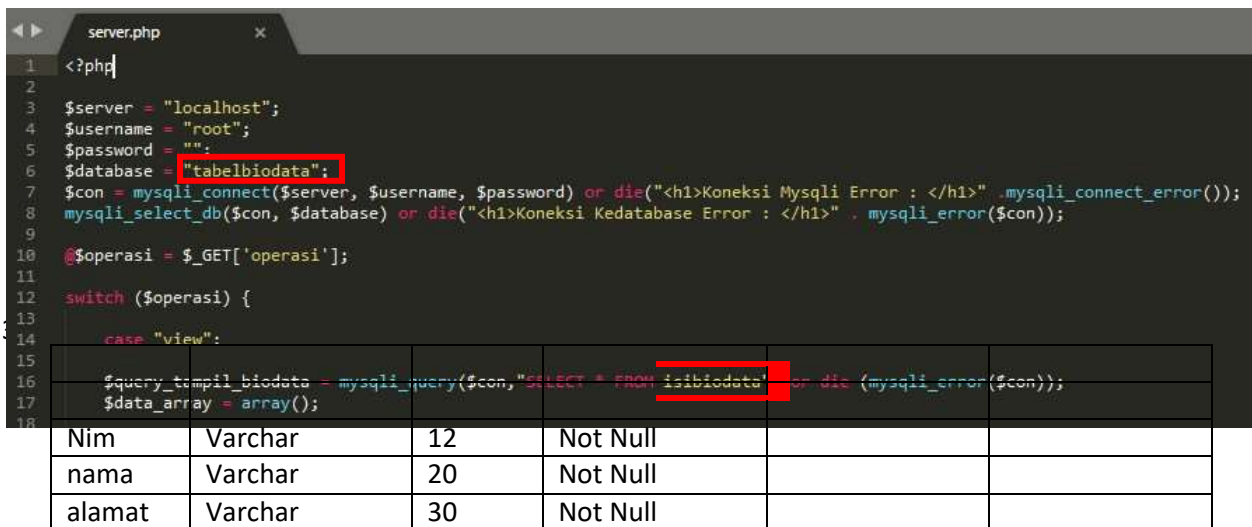
1. Disain database
2. Pengkodean PHP(web service)
3. Rancangan UI Android(client)

B. Disain Database

Rancangan database sebagai berikut:

1. Nama database : **tabelbiodata**
2. Nama tabel: **isibiodata**

Perlu diperhatikan dalam mendisain database adalah nama database dan nama tabel serta nama field akan berpengaruh pada tahapan yang ke dua, **Pengkodean PHP(web service)**. Contoh penggalan script yang berhubungan dengan penamaan database dan tabel:



```
1 <?php
2
3 $server = "localhost";
4 $username = "root";
5 $password = "";
6 $database = "tabelbiodata";
7 $con = mysqli_connect($server, $username, $password) or die("<h1>Koneksi Mysql Error : </h1>" . mysqli_connect_error());
8 mysqli_select_db($con, $database) or die("<h1>Koneksi Kedatabase Error : </h1>" . mysqli_error($con));
9
10 @$operasi = $_GET['operasi'];
11
12 switch ($operasi) {
13
14     case "view":
15
16         $query_tampil_biodata = mysqli_query($con, "SELECT * FROM isibiodata") or die (mysqli_error($con));
17         $data_array = array();
18
```

Nim	Varchar	12	Not Null		
nama	Varchar	20	Not Null		
alamat	Varchar	30	Not Null		

Pastikan struktur tabel(field) **isibiodata** mengikuti tabel diatas, termasuk format name/field

4. Entry data ke dalam tabel mahasiswa sebagai berikut:

nim	Nama	alamat
12161058	MARISKA AYUDYA	BSD Sektor XIV Blok C1/1, Jl. Letnan Sutopo BSD Serpong
12166174	ALDI TRI	Jl. Kamal Raya No.18, RT.6/RW.3, Cengkareng Barat

Catatan: pada tabel entry data ke tabel isibiodata tidak ada data id, karena pada struktur tabel mahasiswa terdapat id(auto_increment) artinya data id akan secara otomatis terisi setiap entry data.

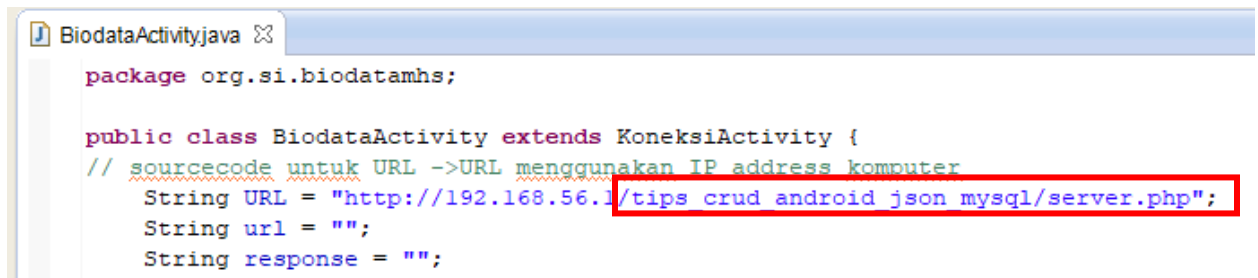
C. Pengkodean PHP(web service)

Database sudah selesai, selanjutnya adalah membuat script server side yang bertugas sebagai web service. Koq kita butuh script server side, Kita kan belajar Android (Java) ? ... Sebenarnya Android tidak dapat berkomunikasi langsung dengan MySQL untuk memanipulasi database. Untuk itu, kita membutuhkan script server side misalnya ASP, JSP atau PHP untuk bisa menjadi “jembatan” antara Android dengan MySQL. Jadi, Android hanya memberikan perintah (request) ke PHP dan PHP lah yang akan melaksanakan query ke MySQL. Dalam studi kasus minggu ke 6,7 dibutuhkan rancangan/script PHP(web service):

Script	Keterangan
server.php	Script dalam file server.php akan dijelaskan dalam pembelajaran berlangsung melalui fasilitas forum

Catatan:

1. file **server.php** di atas diletakkan dalam struktur direktori >>> **C:/xampp/htdocs/tips_crud_android_json_mysql/ server.php.**
2. Struktur penyimpanan dan penamaannya file tersebut akan berpengaruh pada pengkodean/script pada tahap ke 3 (**Rancangan UI Android(client)**). Sample script yang berhubungan sebagai berikut:



```

BiodataActivity.java
package org.si.biodatamhs;

public class BiodataActivity extends KoneksiActivity {
    // sourcecode untuk URL ->URL menggunakan IP address komputer
    String URL = "http://192.168.56.1/tips crud android json mysql/server.php";
    String url = "";
    String response = "";

```

Pengkodean file **server.php**

Penjelasan script tersebut dilakukan pada saat pembelajaran melalui fasilitas forum pada e-learning. Perhatikan script dibawah sudah mengadopsi unit kompetensi dibidang programmer. Melalui pengalaman belajar pada web programming I, II dan III tentu mahasiswa sudah familiar dengan script tersebut, termasuk tanda kutip dua(") dan kutip satu(')

```
server.php
1  <?php
2
3  $server = "localhost";
4  $username = "root";
5  $password = "";
6  $database = "tabelbiodata";
7  $con = mysqli_connect($server, $username, $password) or die("<h1>Koneksi Mysql Error : </h1>" . mysqli_connect_error());
8  mysqli_select_db($con, $database) or die("<h1>Koneksi Kedatabase Error : </h1>" . mysqli_error($con));
9
10 @operasi = $_GET['operasi'];
11
12 switch ($operasi) {
13
14     case "view":
15
16         $query_tampil_biodata = mysqli_query($con, "SELECT * FROM isibiodata") or die (mysqli_error($con));
17         $data_array = array();
18
19         while ($data = mysqli_fetch_assoc($query_tampil_biodata)) {
20             $data_array[]=$data;
21         }
22         echo json_encode($data_array);
23
24         break;
25
26     case "insert":
27         @$nim = $_GET['nim'];
28         @$nama = $_GET['nama'];
29         @$alamat = $_GET['alamat'];
30
31         $query_insert_data = mysqli_query($con, "INSERT INTO isibiodata (nim,nama,alamat) VALUES('$_nim','$nama','$alamat')");
32
33         if ($query_insert_data) {
34             echo "Data Berhasil Disimpan";
35         }
36         else {
37             echo "Maaf Insert Ke Dalam Database Error" . mysqli_error($con);
38         }
39
40         break;
```

```

server.php
41
42     case "get_biodata_by_id":
43         @$id =(int)$_GET['id'];
44         $query_tampil_biodata = mysqli_query($con, "SELECT * FROM isibiodata WHERE id='$id'") or die (mysqli_error($con));
45         $data_array = array();
46         $data_array = mysqli_fetch_assoc($query_tampil_biodata);
47         echo "[" . json_encode ($data_array) . "]";
48         break;
49
50     case "update":
51         @$nim = $_GET['nim'];
52         @$nama = $_GET['nama'];
53         @$alamat = $_GET['alamat'];
54         @$id = $_GET['id'];
55
56         $query_update_biodata = mysqli_query($con, "UPDATE isibiodata SET nim = '$nim', nama='$nama', alamat='$alamat' WHERE id='$id'");
57
58         if ($query_update_biodata) {
59             echo " Update Data Berhasil ";
60         }
61         else {
62             echo mysqli_error($con);
63         }
64         break;
65
66     case "delete":
67         @$id = $_GET['id'];
68         $query_delete_biodata = mysqli_query($con, "DELETE FROM isibiodata WHERE id='$id'");
69         if ($query_delete_biodata) {
70             echo "Data Berhasil Dihapus";
71         }
72         else {
73             echo mysqli_error($con);
74         }
75         break;
76
77     default:
78         break;
79 }
80 }

```

Untuk menguji hasil pengkodean tersebut, tugas mahasiswa adalah melakukan uji menggunakan *postman* seperti praktik pada pertemuan ke 5.

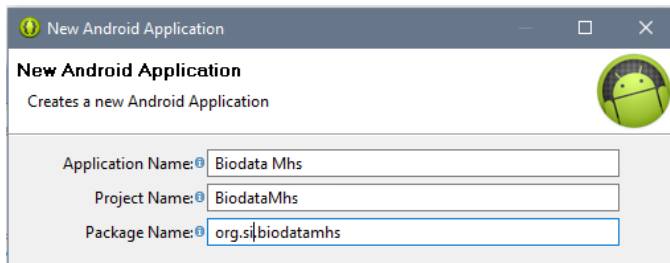
1. GET
2. POST
3. PUT
4. DEL

D. Rancangan UI Android(*client*)

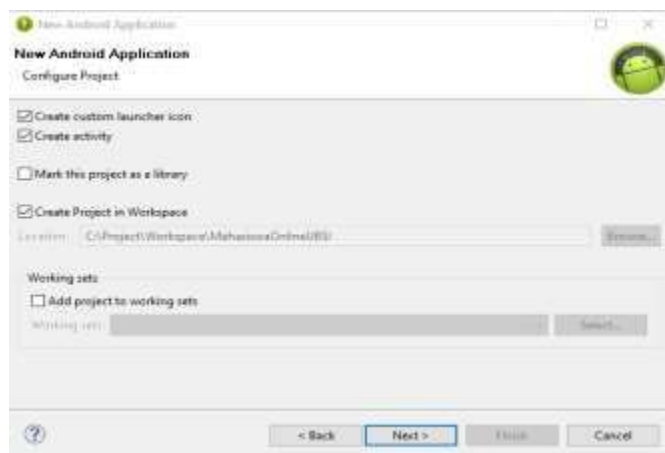
Untuk membuat aplikasi Android menggunakan Eclipse langkah-langkah yang dilakukan adalah sebagai berikut:

1. Buatlah project android dengan data berikut:

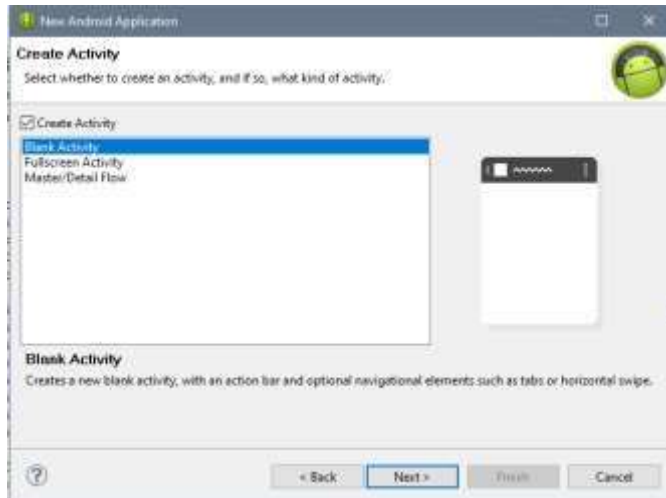
Application Name : Biodata Mhs
Project Name : BiodataMhs
Package : org.si.biodatamhs



2. Klik next, pada tampilan configure project klik next (di awal sudah diatur lokasi penyimpanan project)-Next

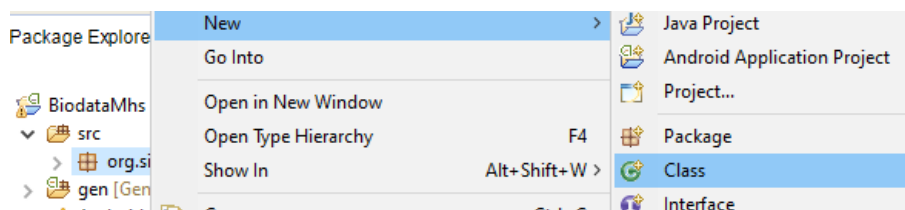


3. Pilih icon yang akan digunakan, Next
4. Pada create Activity pilih Blank Activity-Next

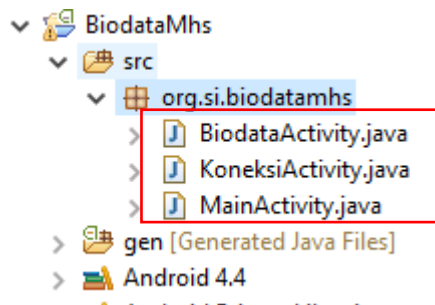


5. Langkah berikutnya adalah penamaan Activity, klik finish (pada kasus ini tidak merubah nama Activity nya)
6. Pada lokasi **src** dan package **org.si.biodatamhs**, tambahkan class **KoneksiActivity.java**, dan **BiodataActivity.java** . caranya adalah sebagai berikut:

- a) Klik kanan pada package **org.si.biodatamhs** pilih **New-Class**
- b) Beri nama class tersebut dengan **KoneksiActivity.java** (.java tidak perlu ditulis)-**finish**



- c) Lakukan hal sama untuk membuat class **BiodataActivity.java**, sehingga terdapat 3 class pada package **org.si.biodatamhs**



7. Sebelum kita mulai membuat code program di class-class tersebut, pastikan aplikasi sudah memiliki izin untuk mengakses internet, caranya adalah sebagai berikut:
 - a) Buka file **AndroidManifest.xml**, tambahkan script berikut:

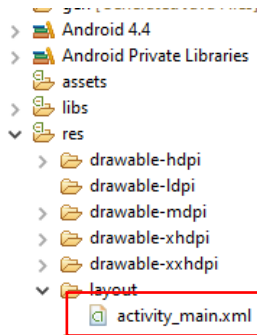

```
<uses-permission android:name="android.permission.INTERNET"/>
```

```

<uses-sdk
    android:minSdkVersion="11"
    android:targetSdkVersion="18" />
<uses-permission android:name="android.permission.INTERNET"/>

```

8. Selanjutnya kita akan menambahkan script pada file **activity_main.xml** pada lokasi **res-layout**



activity_main.xml kita gunakan untuk menampilkan hasil dari biodata. Berikut script program pada

activity_main.xml:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/buttonTambahBiodata"
        android:layout_width="186dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Tambah Biodata"/>

    <HorizontalScrollView
        android:id="@+id/horizontalScrollView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ScrollView
            android:id="@+id/verticalScrollView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TableLayout
                android:id="@+id/tableBiodata"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="80dp">

                </TableLayout>
            </ScrollView>
        </HorizontalScrollView>
    </LinearLayout>

```

9. Berikutnya kita tambahkan script pada class KoneksiActivity.java, berikut script program lengkapnya:

```
package org.si.biodatamhs;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.*;
import java.net.*;

public class KoneksiActivity {

    //mendapatkan nilai bit yg diperlukan
    public String call(String url){
        int BUFFER_SIZE = 2000;
        InputStream in = null;
        try{
            in = openHttpConnection(url);
        } catch (IOException e) {
            e.printStackTrace();
            return "";
        }
        //membaca nilai bit menjadi nilai karakter
        InputStreamReader inputStreamReader = new InputStreamReader(in);
        int charRead;
        String string = "";
        char[] inputBuffer = new char[BUFFER_SIZE];
        try{
            while ((charRead = inputStreamReader.read(inputBuffer)) > 0) {
                String readString = String.valueOf(inputBuffer,0,charRead);
                string += readString;
                inputBuffer = new char[BUFFER_SIZE];
            }
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
            return "";
        }
        return string;
    }

    //membuka URL dan meminta respon dari input streamreader
    private InputStream openHttpConnection(String url) throws IOException {
        InputStream in = null;
        int response = -1;
        URL url1 = new URL(url);
        URLConnection conn = url1.openConnection();
        if(!(conn instanceof HttpURLConnection)) throw new IOException("Not An Http Connection");
        try{
            HttpURLConnection httpURLConnection = (HttpURLConnection) conn;
            httpURLConnection.setAllowUserInteraction(false);
            httpURLConnection.setInstanceFollowRedirects(true);
            httpURLConnection.setRequestMethod("GET");
            httpURLConnection.connect();
            response = httpURLConnection.getResponseCode();
            if(response == HttpURLConnection.HTTP_OK) {
                in = httpURLConnection.getInputStream();
            }
        } catch (Exception e) {
            // TODO: handle exception
            throw new IOException("Error Connecting");
        }
        return in;
    }
}
```

10. Script berikutnya yang harus kita buat adalah BiodataActivity.java. script ini berfungsi untuk menjalankan perintah-perintah SQL (Insert, delete, update, select, dan akses database melalui IP), scriptnya lengkapnya adalah sebagai berikut:

```
package org.si.biodatamhs;

public class BiodataActivity extends KoneksiActivity {
    // sourcecode untuk URL ->URL menggunakan IP address default eclipse
    String URL = "http://10.0.2.2/tips_crud_android_json_mysql/server.php";
}
```

```

String url = "";
String response = "";

//menampilkan biodata dari database
public String tampilBiodata() {
    try{
        url = URL + "?operasi=view";
        System.out.println("URL Tampil Biodata : " + url);
        response = call(url);
    }
    catch(Exception e) {
    }
    return response;
}

//memasukan biodata baru ke dlam database
public String insertBiodata(String nim,String nama,String alamat) {
    try{
        url = URL + "?operasi=insert&nim=" + nim + "&nama=" + nama + "&alamat=" + alamat;
        System.out.println("URL Insert Biodata : " + url);
        response = call(url);
    }
    catch (Exception e){
    }
    return response;
}

//melihat biodata berdasarkan ID
public String getBiodataById (int id) {
    try{
        url=URL + "?operasi=get_biodata_by_id&id=" + id;
        System.out.println("URL Insert Biodata : " + url);
        response = call(url);
    }
    catch(Exception e) {
    }
    return response;
}

//mengubah isi biodata
public String updateBiodata(String id, String nim, String nama, String alamat) {
    try{
        url=URL + "?operasi=update&id=" + id + "&nim=" + nim + "&nama=" + nama + "&alamat=" +
alamat;
        System.out.println("URL Update Biodata : " + url);
        response = call(url);
    }

    catch(Exception e){
    }
    return response;
}

//coding hapus
public String deleteBiodata (int id) {
    try{
        url = URL + "?operasi=delete&id=" + id;
        System.out.println("URL Hapus Biodata : " + url);
        response = call(url);
    }
    catch(Exception e){
    }
    return response;
}
}

```

11. Script terakhir yang harus kita buat adalah pada class MainActivity.java. script ini sangat penting karena merupakan script untuk menampilkan hasil dari pekerjaan kita pada emulator. Karena itu script ini cukup banyak. Berikut script program dari MainActivity.java

```

package org.si.biodatamhs;

import java.util.ArrayList;

```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
import android.os.*;
import android.graphics.*;
import android.app.ActionBar.LayoutParams;
import android.content.DialogInterface;

public class MainActivity extends Activity implements OnClickListener {

    BiodataActivity biodataActivity = new BiodataActivity();
    TableLayout tableLayout;
    Button buttonTambahBiodata;
    ArrayList<Button>buttonEdit = new ArrayList<Button>();
    ArrayList<Button>buttonDelete = new ArrayList<Button>();
    JSONArray arrayBiodata;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Jika SDK Android diatas API Ver.9
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
                .permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        // Mendapatkan data widget dari XML Activity melalui ID tableLayout =
        (TableLayout) findViewById(R.id.tableBiodata); buttonTambahBiodata =
        (Button) findViewById(R.id.buttonTambahBiodata);
        buttonTambahBiodata.setOnClickListener(this);

        //menambah baris untuk tabel
        TableRow barisTabel = new TableRow(this);
        barisTabel.setBackgroundColor(Color.CYAN);

        // Menambahkan tampilan teks untuk judul pada tabel
        TextView viewHeaderId = new TextView(this);
        TextView viewHeaderNim = new TextView(this);
        TextView viewHeaderNama = new TextView(this);
        TextView viewHeaderAlamat = new TextView(this);
        TextView viewHeaderAction = new TextView(this);

        viewHeaderId.setText("ID");
        viewHeaderNim.setText("NIM");
        viewHeaderNama.setText("Nama");
        viewHeaderAlamat.setText("Alamat");
        viewHeaderAction.setText("Action");

        viewHeaderId.setPadding(5, 1, 5, 1);
        viewHeaderNim.setPadding(5, 1, 5, 1);
        viewHeaderNama.setPadding(5, 1, 5, 1);
        viewHeaderAlamat.setPadding(5, 1, 5, 1);
        viewHeaderAction.setPadding(5, 1, 5, 1);

        // Menampilkan tampilan TextView ke dalam tabel
        barisTabel.addView(viewHeaderId);
        barisTabel.addView(viewHeaderNim);
        barisTabel.addView(viewHeaderNama);
        barisTabel.addView(viewHeaderAlamat);
        barisTabel.addView(viewHeaderAction);
    }
}

```

```

        // Menyusun ukuran dari tabel
        tableLayout.addView(barisTabel, new
TableLayout.LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));

        try {
            // Mengubah data dari BiodataActivity yang berupa String menjadi array
            arrayBiodata = new JSONArray(biodataActivity.tampilBiodata());
            for (int i = 0; i < arrayBiodata.length(); i++) {
                JSONObject jsonChildNode = arrayBiodata.getJSONObject(i);
                String nim = jsonChildNode.optString("nim");
                String nama = jsonChildNode.optString("nama");
                String alamat = jsonChildNode.optString("alamat");
                String id = jsonChildNode.optString("id");

                System.out.println("NIM : " + nim );
                System.out.println("Nama : " + nama );
                System.out.println("Alamat : " + alamat);
                System.out.println("ID : " + id);

                barisTabel = new TableRow(this);

                // Memberi warna pada baris tabel
                if (i % 2 == 0) {
                    barisTabel.setBackgroundColor(Color.LTGRAY);
                }

                TextView viewId = new TextView(this);
                viewId.setText(id);
                viewId.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewId);

                TextView viewNim = new TextView(this);
                viewNim.setText(nim);
                viewNim.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewNim);

                TextView viewNama = new TextView(this);
                viewNama.setText(nama);
                viewNama.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewNama);

                TextView viewAlamat = new TextView(this);
                viewAlamat.setText(alamat);
                viewAlamat.setPadding(5, 1, 5, 1);
                barisTabel.addView(viewAlamat);

                // Menambahkan button Edit buttonEdit.add(i,
                new Button(this));
                buttonEdit.get(i).setId(Integer.parseInt(id));
                buttonEdit.get(i).setTag("Edit");
                buttonEdit.get(i).setText("Edit");
                buttonEdit.get(i).setOnClickListener(this);
                barisTabel.addView(buttonEdit.get(i));

                // Menambahkan tombol Delete buttonDelete.add(i,
                new Button(this));
                buttonDelete.get(i).setId(Integer.parseInt(id));
                buttonDelete.get(i).setTag("Delete");
                buttonDelete.get(i).setText("Delete");
                buttonDelete.get(i).setOnClickListener(this);
                barisTabel.addView(buttonDelete.get(i));

                tableLayout.addView(barisTabel, new TableLayout.LayoutParams
                    (LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

public void onClick (View view) {
    if (view.getId() == R.id.buttonTambahBiodata) {
        tambahBiodata();
    }
    else {
        for (int i= 0; i < buttonEdit.size(); i++) {
            // Jika ingin mengedit data pada biodata
            if (view.getId() == buttonEdit.get(i).getId() &&
view.getTag().toString().trim().equals("Edit")) {
                Toast.makeText(MainActivity.this, "Edit : " + buttonEdit.get(i).getId(),
Toast.LENGTH_SHORT).show();
                int id = buttonEdit.get(i).getId();
                getDataById(id);
            }
            // Menghapus data di Tabel
            else if (view.getId() == buttonDelete.get(i).getId() &&
view.getTag().toString().trim().equals("Delete")) {
                Toast.makeText(MainActivity.this, "Delete : " +
buttonDelete.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonDelete.get(i).getId();
                deleteBiodata(id);
            }
        }
    }
}

public void deleteBiodata (int id) {
    biodataActivity.deleteBiodata(id);
    finish();
    startActivity(getIntent());
}

// Mendapatkan Biodata melalui ID
public void getDataById (int id) {
    String namaEdit = null, alamatEdit = null, nimEdit = null;
    JSONArray arrayPersonal;

    try {
        arrayPersonal = new JSONArray(biodataActivity.getBiodataById(id));
        for (int i = 0; i < arrayPersonal.length(); i++) {
            JSONObject jsonChildNode = arrayPersonal.getJSONObject(i);
            nimEdit = jsonChildNode.optString("nim");
            namaEdit = jsonChildNode.optString("nama");
            alamatEdit = jsonChildNode.optString("alamat");
        }
    }
    catch (JSONException e) {
        e.printStackTrace();
    }

    LinearLayout layoutInput = new LinearLayout(this);
    layoutInput.setOrientation(LinearLayout.VERTICAL);

    // Membuat id tersembunyi pada AlertDialog
    final TextView viewId = new TextView(this);
    viewId.setText(String.valueOf(id));
    viewId.setTextColor(Color.TRANSPARENT);
    layoutInput.addView(viewId);

    final EditText editNim = new EditText(this);
    editNim.setText(nimEdit);
    layoutInput.addView(editNim);

    final EditText editNama = new EditText(this);
    editNama.setText(namaEdit);
    layoutInput.addView(editNama);

    final EditText editAlamat = new EditText(this);
    editAlamat.setText(alamatEdit);
    layoutInput.addView(editAlamat);
}

```

```

        // Membuat AlertDialog untuk mengubah data di Biodata
        AlertDialog.Builder builderEditBiodata = new AlertDialog.Builder(this);
        //builderEditBiodata.setIcon(R.drawable.webse);
        builderEditBiodata.setTitle("Update Biodata");
        builderEditBiodata.setView(layoutInput);
        builderEditBiodata.setPositiveButton("Update", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String nim = editNim.getText().toString();
                String nama = editNama.getText().toString();
                String alamat = editAlamat.getText().toString();
                System.out.println("NIM : " + nim + "Nama : " + nama + "Alamat : " + alamat);

                String laporan = biodataActivity.updateBiodata(viewId.getText().toString(),
                editNim.getText().toString(), editNama.getText().toString(),
                editAlamat.getText().toString());

                Toast.makeText(MainActivity.this, laporan, Toast.LENGTH_SHORT).show();
                finish();
                startActivity(getIntent());
            }
        });

        // Jika tidak ingin mengubah data pada Biodata
        builderEditBiodata.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });

        builderEditBiodata.show();
    }

    public void tambahBiodata() {
        LinearLayout layoutInput = new LinearLayout(this);
        layoutInput.setOrientation(LinearLayout.VERTICAL);

        final EditText editNim = new EditText(this);
        editNim.setHint("NIM");
        layoutInput.addView(editNim);

        final EditText editNama = new EditText(this);
        editNama.setHint("Nama");
        layoutInput.addView(editNama);

        final EditText editAlamat = new EditText(this);
        editAlamat.setHint("Alamat");
        layoutInput.addView(editAlamat);

        // Membuat AlertDialog untuk menambahkan data pada Biodata
        AlertDialog.Builder builderInsertBiodata= new AlertDialog.Builder(this);
        //builderInsertBiodata.setIcon(R.drawable.webse);
        builderInsertBiodata.setTitle("Insert Biodata");
        builderInsertBiodata.setView(layoutInput);
        builderInsertBiodata.setPositiveButton("Insert", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String nim = editNim.getText().toString();
                String nama = editNama.getText().toString();
                String alamat = editAlamat.getText().toString();
                System.out.println("NIM : " + nim + "Nama : " + nama + "Alamat : " + alamat);

                String laporan = biodataActivity.insertBiodata(nim, nama, alamat);
                Toast.makeText(MainActivity.this, laporan, Toast.LENGTH_SHORT).show();

                finish();
            }
        });
    }

```

```

        startActivity(getIntent());
    }
});

builderInsertBiodata.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
builderInsertBiodata.show();

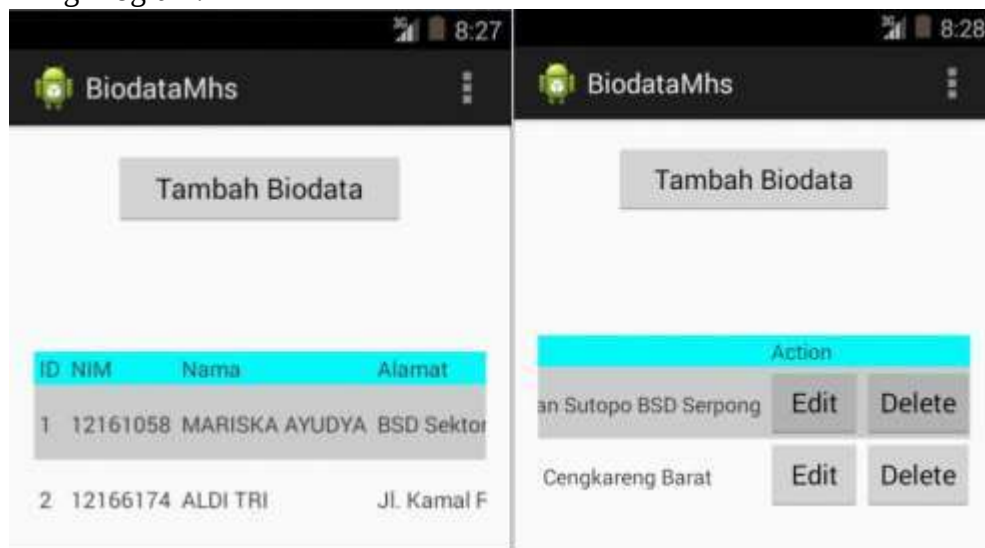
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
}

```

12. untuk menjalankan hasil pekerjaan yang telah kita kerjakan, klik kanan pada project – Android Tools – Fix Project Properties. Jika tidak terjadi kesalahan klik kanan kembali project – Build Project. Dan terakhir klik kanan project – Run As – 1 Android Application (pastikan Android Virtual Device Manager sudah dibuat)

13. Hasil running Program:



Tampilan view data

The screenshot shows the 'BiodataMhs' application interface. At the top, there is a status bar with '3G' signal, battery level, and the time '8:30'. Below the status bar is a dark header with the app icon and the text 'BiodataMhs'. A button labeled 'Tambah Biodata' is centered below the header. A white dialog box titled 'Insert Biodata' is displayed in the center. It contains three text input fields labeled 'NIM', 'Nama', and 'Alamat'. At the bottom of the dialog are two buttons: 'Cancel' on the left and 'Insert' on the right.

Tampilan Insert Data

The screenshot shows the 'BiodataMhs' application interface. At the top, there is a status bar with '3G' signal, battery level, and the time '8:31'. Below the status bar is a dark header with the app icon and the text 'BiodataMhs'. A button labeled 'Tambah Biodata' is centered below the header. A white dialog box titled 'Update Biodata' is displayed in the center. It contains three text input fields. The first field, labeled 'NIM', contains the value '12166174'. The second field, labeled 'Nama', contains the value 'ALDI TRI'. The third field, labeled 'Alamat', contains the value 'Jl. Kamal Raya No.18, RT.6/ RW.3, Cengkareng Barat'. At the bottom of the dialog are two buttons: 'Cancel' on the left and 'Update' on the right.

Tampilan Update Data

Pengaplikasian Project pada Smartphone

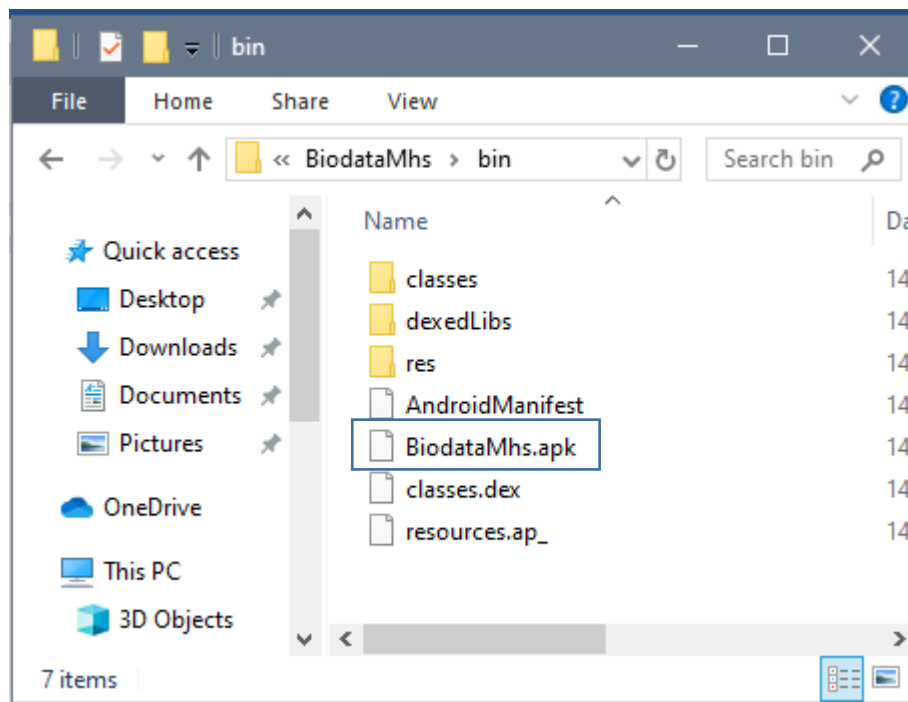
Salah satu cara menampilkan hasil project android kita tanpa menggunakan emulator adalah melalui smartphone kita. Langkah mudahnya adalah kita cukup mengatur IP address saja. Langkah- langkahnya adalah sebagai berikut:

1. Lakukan Share wi-fi melalui Smartphone. Kemudian koneksikan wi-fi pada laptop ke smartphone tersebut. Laptop akan mendapat Ip address. Sebagai contoh IP address laptop adalah **192.168.43.196. Cek Ip laptop masing-masing dan ikuti langkah 2**

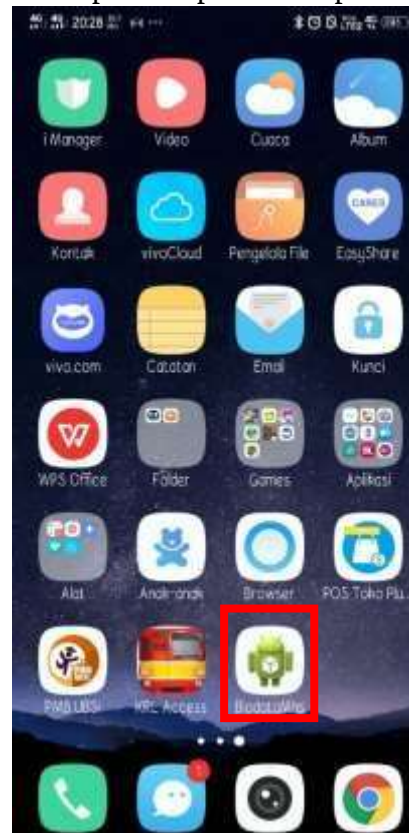
2. Buka class BiodataActivity.java, ubah IP address menjadi IP address yang ada pada laptop

```
package org.si.biodatamhs;  
  
public class BiodataActivity extends KoneksiActivity {  
    // sourcecode untuk URL ->URL menggunakan IP address komputer  
    String URL = "http://192.168.43.196/tips_crud_android_json_mysql/server.php";  
    String url = "";  
    String response = "";
```

3. Pilih menu Project – Clean (untuk menghilangkan pengaturan sebelumnya)
4. klik kanan pada project – Android Tools – Fix Project Properties
5. Pilih menu Project – Build Project
6. klik kanan project – Run As – 1 Android Application (pastikan program running)
7. Buka folder tempat anda menyimpan project, sebagai contoh saya menyimpan project di C:\851_TWS\workspace\BiodataMhs, dalam folder bin terdapat file BiodataMhs.apk, silakan copykan file tersebut ke Smartphone.
Kemudian Install

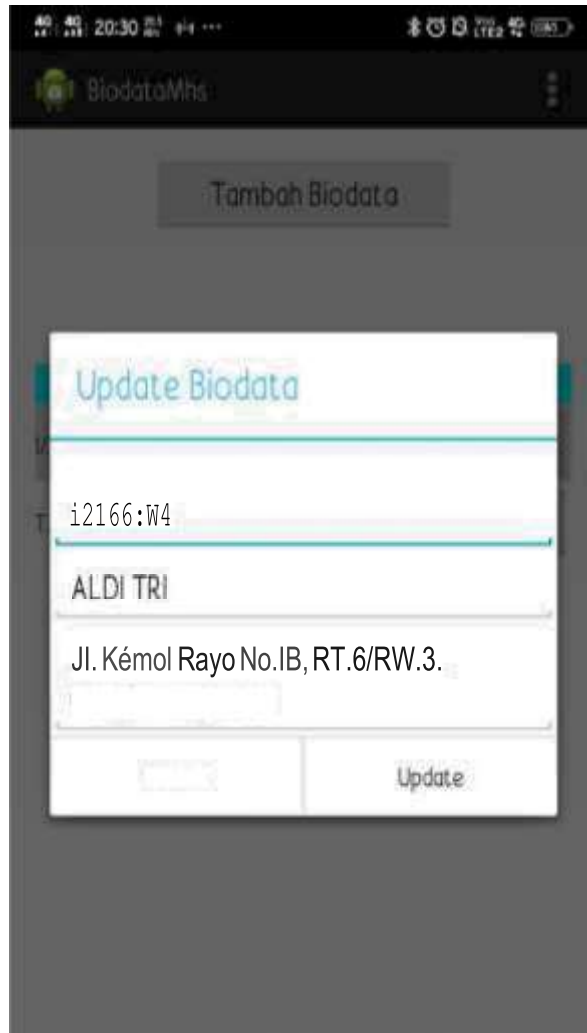


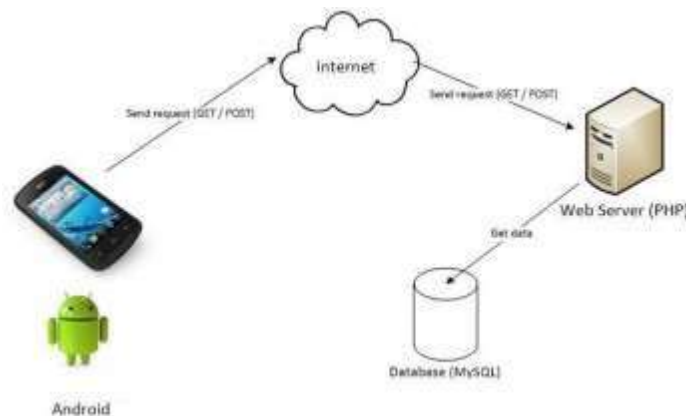
8. Jika sudah terinstall maka akan tampil icon pada smartphone



9. Hasil running aplikasi di smartphone:







Sebagai gambaran aplikasi web service yang akan dipraktikan diilustrasikan seperti gambar di atas. Ponsel Android akan melakukan request (get/post) ke server melalui internet. Cara ini juga sama jika ingin aplikasi berada pada localhost. Selanjutnya web server (dalam hal ini PHP), akan memproses request dari Android dan akan melakukan query ke database (MySQL). Untuk menghasilkan konsep di atas dibutuhkan 3 langkah:

1. Disain database
2. Pengkodean PHP(web service)
3. Rancangan UI Android(client)

B. Disain Database

Rancangan database sebagai berikut:

1. Nama database : **tabelbiodata**
2. Nama tabel: **isibiodata**

Perlu diperhatikan dalam mendisain database adalah nama database dan nama tabel serta nama field akan berpengaruh pada tahapan yang ke dua, **Pengkodean PHP(web service)**. Contoh penggalan script yang berhubungan dengan penamaan database dan tabel:

```

server.php
1 <?php
2
3 $server = "localhost";
4 $username = "root";
5 $password = "";
6 $database = "tabelbiodata";
7 $con = mysqli_connect($server, $username, $password) or die("<h1>Koneksi Mysql Error : </h1>" . mysqli_connect_error());
8 mysqli_select_db($con, $database) or die("<h1>Koneksi Kedatabase Error : </h1>" . mysqli_error($con));
9
10 @$operasi = $_GET['operasi'];
11
12 switch ($operasi) {
13
14     case "view":
15
16         $query_tampil biodata = mysqli_query($con, "SELECT * FROM isibiodata" or die (mysqli_error($con));
17         $data_array = array();
18
19
20

```

Nim	Varchar	12	Not Null		
nama	Varchar	20	Not Null		
alamat	Varchar	30	Not Null		

3. Struktur tabel sebagai berikut:

Name	Type	Length	NULL/Note Null	Key	A I
Id	Int	11	Not Null	Primary	Auto_increment

Pastikan struktur tabel(field) **isibiodata** mengikuti tabel diatas, termasuk format name/field

4. Entry data ke dalam tabel mahasiswa sebagai berikut:

nim	Nama	alamat
12161058	MARISKA AYUDYA	BSD Sektor XIV Blok C1/1, Jl. Letnan Sutopo BSD Serpong
12166174	ALDI TRI	Jl. Kamal Raya No.18, RT.6/RW.3, Cengkareng Barat

Catatan: pada tabel entry data ke tabel isibiodata tidak ada data id, karena pada struktur tabel mahasiswa terdapat id(auto_increment) artinya data id akan secara otomatis terisi setiap entry data.

C. Pengkodean PHP(web service)

Database sudah selesai, selanjutnya adalah membuat script server side yang bertugas sebagai web service. Koq kita butuh script server side, Kita kan belajar Android (Java) ? Sebenarnya Android tidak dapat berkomunikasi langsung dengan MySQL untuk memanipulasi database. Untuk itu, kita membutuhkan script server side misalnya ASP, JSP atau PHP untuk bisa menjadi “jembatan” antara Android dengan MySQL. Jadi, Android hanya memberikan perintah (request) ke PHP dan PHP lah yang akan melaksanakan query ke MySQL. Dalam studi kasus minggu ke 6,7 dibutuhkan rancangan/script PHP(web service):

Script	Keterangan
server.php	Script dalam file server.php akan dijelaskan dalam pembelajaran berlangsung melalui fasilitas forum

Catatan:

1. file **server.php** di atas diletakkan dalam struktur direktori >>> **C:/xampp/htdocs/tips_crud_android_json_mysql/ server.php.**
2. Struktur penyimpanan dan penamaannya file tersebut akan berpengaruh pada pengkodean/script pada tahap ke 3 (**Rancangan UI Android(client)**). Sample script yang berhubungan sebagai berikut:

```

BiodataActivity.java
package org.si.biodatamhs;

public class BiodataActivity extends KoneksiActivity {
    // sourcecode untuk URL ->URL menggunakan IP address komputer
    String URL = "http://192.168.56.1/tips_crud_android_json_mysql/server.php";
    String url = "";
    String response = "";

```

Pengkodean file **server.php**

Penjelasan script tersebut dilakukan pada saat pembelajaran melalui fasilitas forum pada e-learning. Perhatikan sript dibawah sudah mengadopsi unit kompetensi dibidang programmer. Melalui pengalaman belajar pada web programming I, II dan III tentu mahasiswa sudah familiar dengan script tersebut, termasuk tanda kutip dua (“) dan kutip satu(‘)

```
server.php
1  <?php
2
3  $server = "localhost";
4  $username = "root";
5  $password = "";
6  $database = "tabelbiodata";
7  $con = mysqli_connect($server, $username, $password) or die("<h1>Koneksi Mysql Error : </h1>" . mysqli_connect_error());
8  mysqli_select_db($con, $database) or die("<h1>Koneksi Kedatabase Error : </h1>" . mysqli_error($con));
9
10 @operasi = $_GET['operasi'];
11
12 switch ($operasi) {
13
14     case "view":
15
16         $query_tampil_biodata = mysqli_query($con,"SELECT * FROM isibiodata") or die (mysqli_error($con));
17         $data_array = array();
18
19         while ($data = mysqli_fetch_assoc($query_tampil_biodata)) {
20             $data_array[]=$data;
21         }
22         echo json_encode($data_array);
23
24         break;
25
26     case "insert":
27         @$nim = $_GET['nim'];
28         @$nama = $_GET['nama'];
29         @$alamat = $_GET['alamat'];
30
31         $query_insert_data = mysqli_query($con, "INSERT INTO isibiodata (nim,nama,alamat) VALUES('$_nim','$nama','$alamat')");
32
33         if ($query_insert_data) {
34             echo "Data Berhasil Disimpan";
35         }
36         else {
37             echo "Maaf Insert Ke Dalam Database Error" . mysqli_error($con);
38         }
39
40         break;
```

```

server.php
41
42     case "get_biodata_by_id":
43         @$id =(int)$_GET['id'];
44         $query_tampil_biodata = mysqli_query($con, "SELECT * FROM isibiodata WHERE id='$id'") or die (mysqli_error($con));
45         $data_array = array();
46         $data_array = mysqli_fetch_assoc($query_tampil_biodata);
47         echo "[" . json_encode ($data_array) . "]";
48         break;
49
50     case "update":
51         @$nim = $_GET['nim'];
52         @$nama = $_GET['nama'];
53         @$alamat = $_GET['alamat'];
54         @$id = $_GET['id'];
55
56         $query_update_biodata = mysqli_query($con, "UPDATE isibiodata SET nim = '$nim', nama='$nama', alamat='$alamat' WHERE id='$id'");
57
58         if ($query_update_biodata) {
59             echo " Update Data Berhasil ";
60         }
61         else {
62             echo mysqli_error($con);
63         }
64         break;
65
66     case "delete":
67         @$id = $_GET['id'];
68         $query_delete_biodata = mysqli_query($con, "DELETE FROM isibiodata WHERE id='$id'");
69         if ($query_delete_biodata) {
70             echo "Data Berhasil Dihapus";
71         }
72         else {
73             echo mysqli_error($con);
74         }
75         break;
76
77     default:
78         break;
79 }
80 }

```

Untuk menguji hasil pengkodean tersebut, tugas mahasiswa adalah melakukan uji menggunakan *postman* seperti praktik pada pertemuan ke 5.

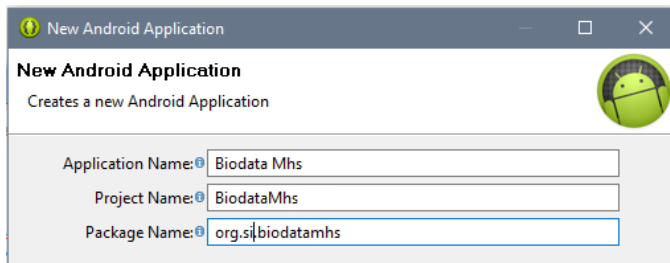
1. GET
2. POST
3. PUT
4. DEL

D. Rancangan UI Android(*client*)

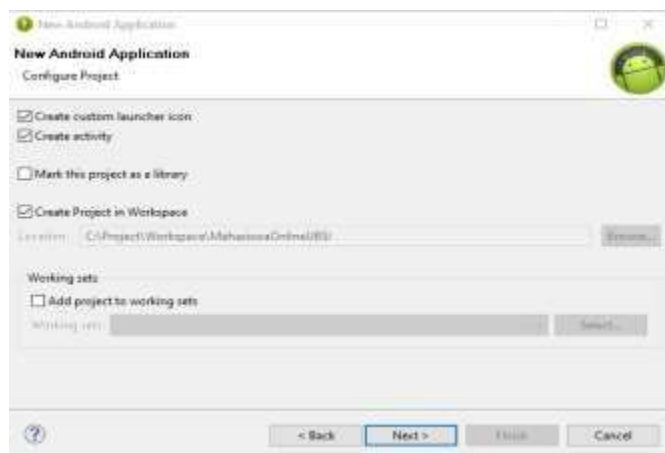
Untuk membuat aplikasi Android menggunakan Eclipse langkah-langkah yang dilakukan adalah sebagai berikut:

1. Buatlah project android dengan data berikut:

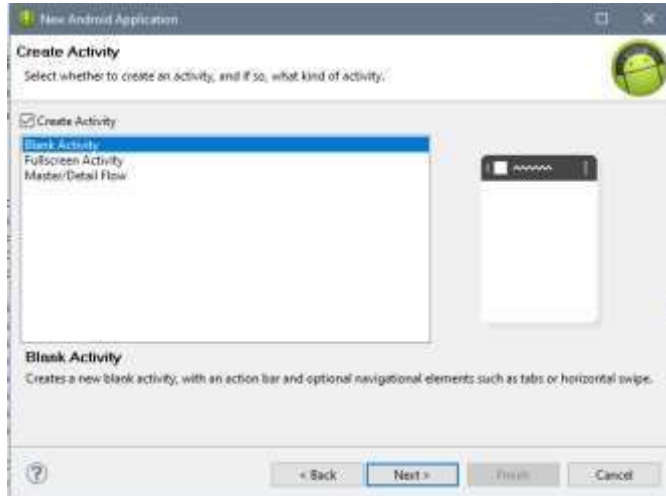
Application Name : Biodata Mhs
Project Name : BiodataMhs
Package : org.si.biodatamhs



2. Klik next, pada tampilan configure project klik next (di awal sudah diatur lokasi penyimpanan project)-Next

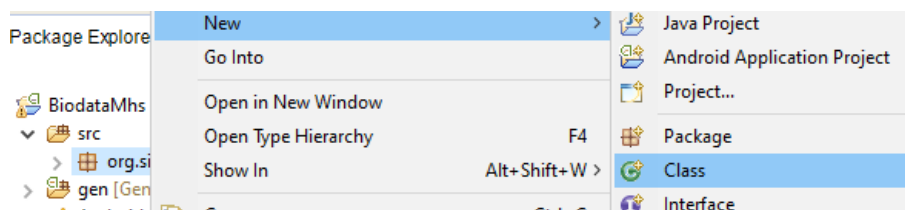


3. Pilih icon yang akan digunakan, Next
4. Pada create Activity pilih Blank Activity-Next

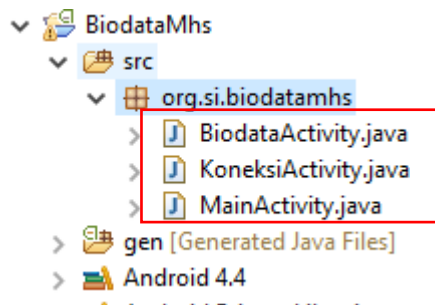


5. Langkah berikutnya adalah penamaan Activity, klik finish (pada kasus ini tidak merubah nama Activity nya)
6. Pada lokasi **src** dan package **org.si.biodatamhs**, tambahkan class **KoneksiActivity.java**, dan **BiodataActivity.java** . caranya adalah sebagai berikut:

- a) Klik kanan pada package **org.si.biodatamhs** pilih **New-Class**
- b) Beri nama class tersebut dengan **KoneksiActivity.java** (.java tidak perlu ditulis)-**finish**



- c) Lakukan hal sama untuk membuat class **BiodataActivity.java**, sehingga terdapat 3 class pada package **org.si.biodatamhs**



7. Sebelum kita mulai membuat code program di class-class tersebut, pastikan aplikasi sudah memiliki izin untuk mengakses internet, caranya adalah sebagai berikut:

- a) Buka file **AndroidManifest.xml**, tambahkan script berikut:

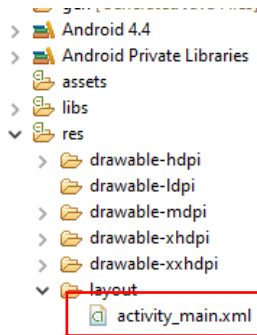
```
<uses-permission android:name="android.permission.INTERNET"/>
```

```

<uses-sdk
    android:minSdkVersion="11"
    android:targetSdkVersion="18" />
<uses-permission android:name="android.permission.INTERNET"/>

```

8. Selanjutnya kita akan menambahkan script pada file **activity_main.xml** pada lokasi **res-layout**



activity_main.xml kita gunakan untuk menampilkan hasil dari biodata. Berikut script program pada

activity_main.xml:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/buttonTambahBiodata"
        android:layout_width="186dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Tambah Biodata"/>

    <HorizontalScrollView
        android:id="@+id/horizontalScrollView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ScrollView
            android:id="@+id/verticalScrollView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TableLayout
                android:id="@+id/tableBiodata"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="80dp">

                </TableLayout>
            </ScrollView>
        </HorizontalScrollView>
    </LinearLayout>

```

9. Berikutnya kita tambahkan script pada class KoneksiActivity.java, berikut script program lengkapnya:

```
package org.si.biodatamhs;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.*;
import java.net.*;

public class KoneksiActivity {

    //mendapatkan nilai bit yg diperlukan
    public String call(String url){
        int BUFFER_SIZE = 2000;
        InputStream in = null;
        try{
            in = openHttpConnection(url);
        } catch (IOException e) {
            e.printStackTrace();
            return "";
        }
        //membaca nilai bit menjadi nilai karakter
        InputStreamReader inputStreamReader = new InputStreamReader(in);
        int charRead;
        String string = "";
        char[] inputBuffer = new char[BUFFER_SIZE];
        try{
            while ((charRead = inputStreamReader.read(inputBuffer)) > 0) {
                String readString = String.valueOf(inputBuffer,0,charRead);
                string += readString;
                inputBuffer = new char[BUFFER_SIZE];
            }
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
            return "";
        }
        return string;
    }

    //membuka URL dan meminta respon dari input streamreader
    private InputStream openHttpConnection(String url) throws IOException {
        InputStream in = null;
        int response = -1;
        URL url1 = new URL(url);
        URLConnection conn = url1.openConnection();
        if(!(conn instanceof HttpURLConnection)) throw new IOException("Not An Http Connection");
        try{
            HttpURLConnection httpURLConnection = (HttpURLConnection) conn;
            httpURLConnection.setAllowUserInteraction(false);
            httpURLConnection.setInstanceFollowRedirects(true);
            httpURLConnection.setRequestMethod("GET");
            httpURLConnection.connect();
            response = httpURLConnection.getResponseCode();
            if(response == HttpURLConnection.HTTP_OK) {
                in = httpURLConnection.getInputStream();
            }
        } catch (Exception e) {
            // TODO: handle exception
            throw new IOException("Error Connecting");
        }
        return in;
    }
}
```

10. Script berikutnya yang harus kita buat adalah BiodataActivity.java. script ini berfungsi untuk menjalankan perintah-perintah SQL (Insert, delete, update, select, dan akses database melalui IP), scriptnya lengkapnya adalah sebagai berikut:

```
package org.si.biodatamhs;

public class BiodataActivity extends KoneksiActivity {
    // sourcecode untuk URL ->URL menggunakan IP address default eclipse
    String URL = "http://10.0.2.2/tips_crud_android_json_mysql/server.php";
}
```

```

String url = "";
String response = "";

//menampilkan biodata dari database
public String tampilBiodata() {
    try{
        url = URL + "?operasi=view";
        System.out.println("URL Tampil Biodata : " + url);
        response = call(url);
    }
    catch(Exception e) {
    }
    return response;
}

//memasukan biodata baru ke dlam database
public String insertBiodata(String nim,String nama,String alamat) {
    try{
        url = URL + "?operasi=insert&nim=" + nim + "&nama=" + nama + "&alamat=" + alamat;
        System.out.println("URL Insert Biodata : " + url);
        response = call(url);
    }
    catch (Exception e){
    }
    return response;
}

//melihat biodata berdasarkan ID
public String getBiodataById (int id) {
    try{
        url=URL + "?operasi=get_biodata_by_id&id=" + id;
        System.out.println("URL Insert Biodata : " + url);
        response = call(url);
    }
    catch(Exception e) {
    }
    return response;
}

//mengubah isi biodata
public String updateBiodata(String id, String nim, String nama, String alamat) {
    try{
        url=URL + "?operasi=update&id=" + id + "&nim=" + nim + "&nama=" + nama + "&alamat=" +
alamat;
        System.out.println("URL Update Biodata : " + url);
        response = call(url);
    }

    catch(Exception e){
    }
    return response;
}

//coding hapus
public String deleteBiodata (int id) {
    try{
        url = URL + "?operasi=delete&id=" + id;
        System.out.println("URL Hapus Biodata : " + url);
        response = call(url);
    }
    catch(Exception e){
    }
    return response;
}
}

```

11. Script terakhir yang harus kita buat adalah pada class MainActivity.java. script ini sangat penting karena merupakan script untuk menampilkan hasil dari pekerjaan kita pada emulator. Karena itu script ini cukup banyak. Berikut script program dari MainActivity.java

```

package org.si.biodatamhs;

import java.util.ArrayList;

```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
import android.os.*;
import android.graphics.*;
import android.app.ActionBar.LayoutParams;
import android.content.DialogInterface;

public class MainActivity extends Activity implements OnClickListener {

    BiodataActivity biodataActivity = new BiodataActivity();
    TableLayout tableLayout;
    Button buttonTambahBiodata;
    ArrayList<Button>buttonEdit = new ArrayList<Button>();
    ArrayList<Button>buttonDelete = new ArrayList<Button>();
    JSONArray arrayBiodata;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Jika SDK Android diatas API Ver.9
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
                .permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        // Mendapatkan data widget dari XML Activity melalui ID tableLayout =
        (TableLayout) findViewById(R.id.tableBiodata); buttonTambahBiodata =
        (Button) findViewById(R.id.buttonTambahBiodata);
        buttonTambahBiodata.setOnClickListener(this);

        //menambah baris untuk tabel
        TableRow barisTabel = new TableRow(this);
        barisTabel.setBackgroundColor(Color.CYAN);

        // Menambahkan tampilan teks untuk judul pada tabel
        TextView viewHeaderId = new TextView(this);
        TextView viewHeaderNim = new TextView(this);
        TextView viewHeaderNama = new TextView(this);
        TextView viewHeaderAlamat = new TextView(this);
        TextView viewHeaderAction = new TextView(this);

        viewHeaderId.setText("ID");
        viewHeaderNim.setText("NIM");
        viewHeaderNama.setText("Nama");
        viewHeaderAlamat.setText("Alamat");
        viewHeaderAction.setText("Action");

        viewHeaderId.setPadding(5, 1, 5, 1);
        viewHeaderNim.setPadding(5, 1, 5, 1);
        viewHeaderNama.setPadding(5, 1, 5, 1);
        viewHeaderAlamat.setPadding(5, 1, 5, 1);
        viewHeaderAction.setPadding(5, 1, 5, 1);

        // Menampilkan tampilan TextView ke dalam tabel
        barisTabel.addView(viewHeaderId);
        barisTabel.addView(viewHeaderNim);
        barisTabel.addView(viewHeaderNama);
        barisTabel.addView(viewHeaderAlamat);
        barisTabel.addView(viewHeaderAction);
    }
}

```

```

// Menyusun ukuran dari tabel
tableLayout.addView(barisTabel, new
TableLayout.LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));

try {
// Mengubah data dari BiodataActivity yang berupa String menjadi array
arrayBiodata = new JSONArray(biodataActivity.tampilBiodata());
for (int i = 0; i < arrayBiodata.length(); i++) {
JSONObject jsonChildNode = arrayBiodata.getJSONObject(i);
String nim = jsonChildNode.optString("nim");
String nama = jsonChildNode.optString("nama");
String alamat = jsonChildNode.optString("alamat");
String id = jsonChildNode.optString("id");

System.out.println("NIM : " + nim );
System.out.println("Nama : " + nama );
System.out.println("Alamat : " + alamat);
System.out.println("ID : " + id);

barisTabel = new TableRow(this);

// Memberi warna pada baris tabel
if (i % 2 == 0) {
barisTabel.setBackgroundColor(Color.LTGRAY);
}

TextView viewId = new TextView(this);
viewId.setText(id);
viewId.setPadding(5, 1, 5, 1);
barisTabel.addView(viewId);

TextView viewNim = new TextView(this);
viewNim.setText(nim);
viewNim.setPadding(5, 1, 5, 1);
barisTabel.addView(viewNim);

TextView viewNama = new TextView(this);
viewNama.setText(nama);
viewNama.setPadding(5, 1, 5, 1);
barisTabel.addView(viewNama);

TextView viewAlamat = new TextView(this);
viewAlamat.setText(alamat);
viewAlamat.setPadding(5, 1, 5, 1);
barisTabel.addView(viewAlamat);

// Menambahkan button Edit buttonEdit.add(i,
new Button(this));
buttonEdit.get(i).setId(Integer.parseInt(id));
buttonEdit.get(i).setTag("Edit");
buttonEdit.get(i).setText("Edit");
buttonEdit.get(i).setOnClickListener(this);
barisTabel.addView(buttonEdit.get(i));

// Menambahkan tombol Delete buttonDelete.add(i,
new Button(this));
buttonDelete.get(i).setId(Integer.parseInt(id));
buttonDelete.get(i).setTag("Delete");
buttonDelete.get(i).setText("Delete");
buttonDelete.get(i).setOnClickListener(this);
barisTabel.addView(buttonDelete.get(i));

tableLayout.addView(barisTabel, new TableLayout.LayoutParams
(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
}
}
catch (JSONException e) {
e.printStackTrace();
}
}

```

```

public void onClick (View view) {
    if (view.getId() == R.id.buttonTambahBiodata) {
        tambahBiodata();
    }
    else {
        for (int i= 0; i < buttonEdit.size(); i++) {
            // Jika ingin mengedit data pada biodata
            if (view.getId() == buttonEdit.get(i).getId() &&
view.getTag().toString().trim().equals("Edit")) {
                Toast.makeText(MainActivity.this, "Edit : " + buttonEdit.get(i).getId(),
Toast.LENGTH_SHORT).show();
                int id = buttonEdit.get(i).getId();
                getDataById(id);
            }
            // Menghapus data di Tabel
            else if (view.getId() == buttonDelete.get(i).getId() &&
view.getTag().toString().trim().equals("Delete")) {
                Toast.makeText(MainActivity.this, "Delete : " +
buttonDelete.get(i).getId(), Toast.LENGTH_SHORT).show();
                int id = buttonDelete.get(i).getId();
                deleteBiodata(id);
            }
        }
    }
}

public void deleteBiodata (int id) {
    biodataActivity.deleteBiodata(id);
    finish();
    startActivity(getIntent());
}

// Mendapatkan Biodata melalui ID
public void getDataById (int id) {
    String namaEdit = null, alamatEdit = null, nimEdit = null;
    JSONArray arrayPersonal;

    try {
        arrayPersonal = new JSONArray(biodataActivity.getBiodataById(id));
        for (int i = 0; i < arrayPersonal.length(); i++) {
            JSONObject jsonChildNode = arrayPersonal.getJSONObject(i);
            nimEdit = jsonChildNode.optString("nim");
            namaEdit = jsonChildNode.optString("nama");
            alamatEdit = jsonChildNode.optString("alamat");
        }
    }
    catch (JSONException e) {
        e.printStackTrace();
    }

    LinearLayout layoutInput = new LinearLayout(this);
    layoutInput.setOrientation(LinearLayout.VERTICAL);

    // Membuat id tersembunyi pada AlertDialog
    final TextView viewId = new TextView(this);
    viewId.setText(String.valueOf(id));
    viewId.setTextColor(Color.TRANSPARENT);
    layoutInput.addView(viewId);

    final EditText editNim = new EditText(this);
    editNim.setText(nimEdit);
    layoutInput.addView(editNim);

    final EditText editNama = new EditText(this);
    editNama.setText(namaEdit);
    layoutInput.addView(editNama);

    final EditText editAlamat = new EditText(this);
    editAlamat.setText(alamatEdit);
    layoutInput.addView(editAlamat);
}

```

```

        // Membuat AlertDialog untuk mengubah data di Biodata
        AlertDialog.Builder builderEditBiodata = new AlertDialog.Builder(this);
        //builderEditBiodata.setIcon(R.drawable.webse);
        builderEditBiodata.setTitle("Update Biodata");
        builderEditBiodata.setView(layoutInput);
        builderEditBiodata.setPositiveButton("Update", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String nim = editNim.getText().toString();
                String nama = editNama.getText().toString();
                String alamat = editAlamat.getText().toString();
                System.out.println("NIM : " + nim + "Nama : " + nama + "Alamat : " + alamat);

                String laporan = biodataActivity.updateBiodata(viewId.getText().toString(),
                editNim.getText().toString(), editNama.getText().toString(),
                editAlamat.getText().toString());

                Toast.makeText(MainActivity.this, laporan, Toast.LENGTH_SHORT).show();
                finish();
                startActivity(getIntent());
            }
        });

        // Jika tidak ingin mengubah data pada Biodata
        builderEditBiodata.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });

        builderEditBiodata.show();
    }

    public void tambahBiodata() {
        LinearLayout layoutInput = new LinearLayout(this);
        layoutInput.setOrientation(LinearLayout.VERTICAL);

        final EditText editNim = new EditText(this);
        editNim.setHint("NIM");
        layoutInput.addView(editNim);

        final EditText editNama = new EditText(this);
        editNama.setHint("Nama");
        layoutInput.addView(editNama);

        final EditText editAlamat = new EditText(this);
        editAlamat.setHint("Alamat");
        layoutInput.addView(editAlamat);

        // Membuat AlertDialog untuk menambahkan data pada Biodata
        AlertDialog.Builder builderInsertBiodata= new AlertDialog.Builder(this);
        //builderInsertBiodata.setIcon(R.drawable.webse);
        builderInsertBiodata.setTitle("Insert Biodata");
        builderInsertBiodata.setView(layoutInput);
        builderInsertBiodata.setPositiveButton("Insert", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String nim = editNim.getText().toString();
                String nama = editNama.getText().toString();
                String alamat = editAlamat.getText().toString();
                System.out.println("NIM : " + nim + "Nama : " + nama + "Alamat : " + alamat);

                String laporan = biodataActivity.insertBiodata(nim, nama, alamat);
                Toast.makeText(MainActivity.this, laporan, Toast.LENGTH_SHORT).show();

                finish();
            }
        });
    }

```

```

        startActivity(getIntent());
    }
});

builderInsertBiodata.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
builderInsertBiodata.show();

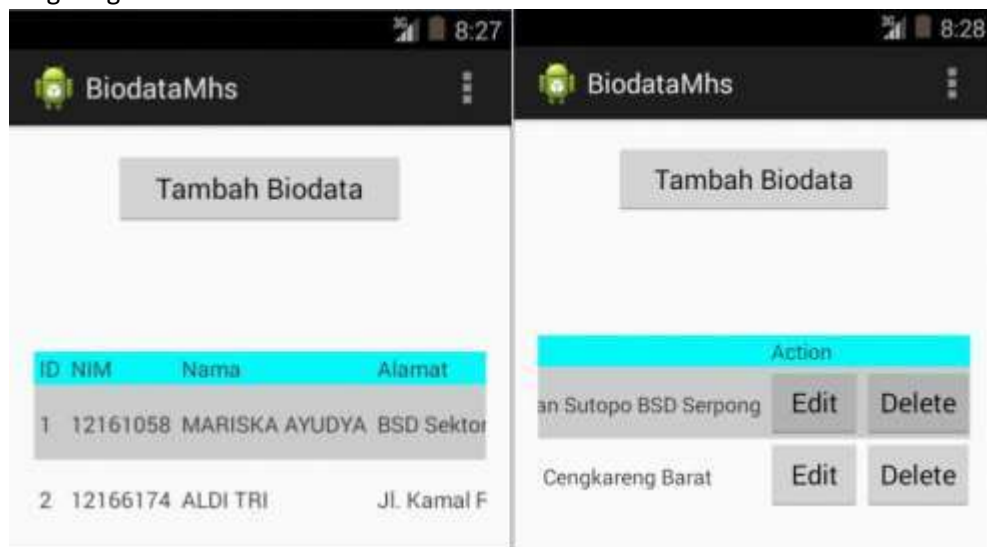
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
}

```

12. untuk menjalankan hasil pekerjaan yang telah kita kerjakan, klik kanan pada project – Android Tools – Fix Project Properties. Jika tidak terjadi kesalahan klik kanan kembali project – Build Project. Dan terakhir klik kanan project – Run As – 1 Android Application (pastikan Android Virtual Device Manager sudah dibuat)

13. Hasil running Program:



Tampilan view data

The screenshot shows the 'BiodataMhs' application interface. At the top, there is a status bar with '3G' signal, battery level, and the time '8:30'. Below the status bar is a header bar with the application icon and the text 'BiodataMhs'. A button labeled 'Tambah Biodata' is visible. The main content area displays a dialog box titled 'Insert Biodata'. This dialog box contains three input fields: 'NIM', 'Nama', and 'Alamat'. At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Insert'.

Tampilan Insert Data

The screenshot shows the 'BiodataMhs' application interface. At the top, there is a status bar with '3G' signal, battery level, and the time '8:31'. Below the status bar is a header bar with the application icon and the text 'BiodataMhs'. A button labeled 'Tambah Biodata' is visible. The main content area displays a dialog box titled 'Update Biodata'. This dialog box contains three input fields with pre-filled data: '12166174', 'ALDI TRI', and 'Jl. Kamal Raya No.18, RT.6/ RW.3, Cengkareng Barat'. At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Update'.

Tampilan Update Data

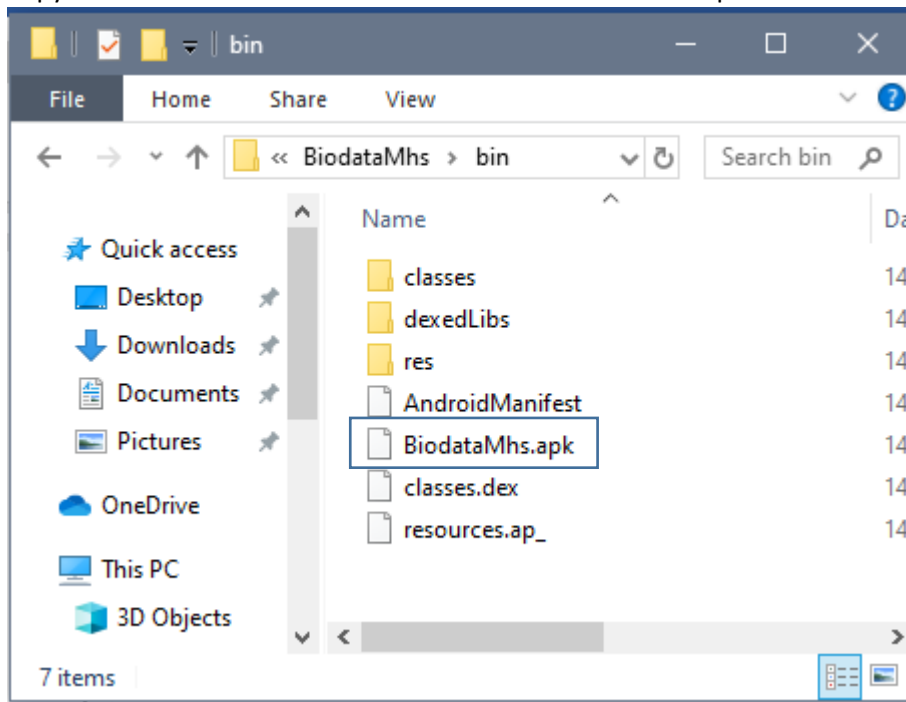
Pengaplikasian Project pada Smartphone

Salah satu cara menampilkan hasil project android kita tanpa menggunakan emulator adalah melalui smartphone kita. Langkah mudahnya adalah kita cukup mengatur IP address saja. Langkah-langkahnya adalah sebagai berikut:

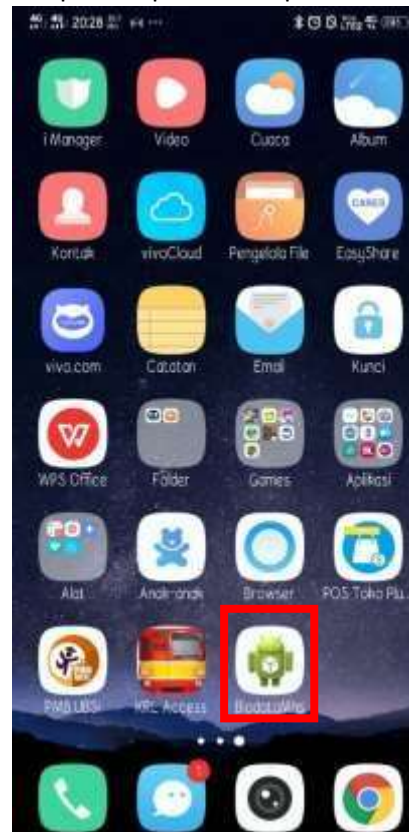
1. Lakukan Share wi-fi melalui Smartphone. Kemudian koneksikan wi-fi pada laptop ke smartphone tersebut. Laptop akan mendapat Ip address. Sebagai contoh IP address laptop adalah **192.168.43.196. Cek Ip laptop masing-masing dan ikuti langkah 2**
2. Buka class BiodataActivity.java, ubah IP address menjadi IP address yang ada pada laptop

```
package org.si.biodatamhs;  
  
public class BiodataActivity extends KoneksiActivity {  
    // sourcecode untuk URL ->URL menggunakan IP address komputer  
    String URL = "http://192.168.43.196/tips_crud_android_json_mysql/server.php";  
    String url = "";  
    String response = "";  
}
```

3. Pilih menu Project – Clean (untuk menghilangkan pengaturan sebelumnya)
4. klik kanan pada project – Android Tools – Fix Project Properties
5. Pilih menu Project – Build Project
6. klik kanan project – Run As – 1 Android Application (pastikan program running)
7. Buka folder tempat anda menyimpan project, sebagai contoh saya menyimpan project di C:\851_TWS\workspace\BiodataMhs, dalam folder bin terdapat file BiodataMhs.apk, silakan copykan file tersebut ke Smartphone. Kemudian Install



8. Jika sudah terinstall maka akan tampil icon pada smartphone



9. Hasil running aplikasi di smartphone:



