BAB II
LANDASAN TEORI

II.1 *Chatting*

Chatting dalam bahasa Indonesia berarti ngobrol atau berbicara dua arah antara satu atau beberapa orang. Di dalam dunia komputer, *chatting* berarti berbicara dengan orang lain dengan menggunakan komputer. Suara yang dihasilkan, biasanya digantikan dengan teks yang diketik. Namun dengan berkembangnya multimedia dengan komputer, *chatting* tidak hanya dengan menggunakan teks, tapi bisa juga dengan menggunakan suara dan video. Lebih sempit lagi, pengertian *chatting* di dalam dunia internet. Di sini pengertian *chatting* adalah menggunakan internet untuk berbicara dengan orang lain.

Menurut jumlah orang yang berbicara, *chatting* dapat dibagi menjadi dua, yaitu *group chat* dan *private chat*. *Group chat* adalah *chatting* yang melibatkan lebih dari dua orang. Biasanya orang-orang ini berkumpul di dalam suatu *chat room* atau *channel*, tempat di mana mereka bisa berinteraksi. Apabila satu orang mengirimkan suatu pesan, maka seluruh orang yang berada di *chat room* atau *channel* tersebut bisa membacanya. Sedangkan *private chat* tidak demikian. *Private chat* hanya melibatkan dua orang. Jadi hanya ada orang tertentu yang dapat membaca pesan kita.

II.2 API

API adalah singkatan dari *Application Program Interface*, yakni serangkaian instruksi dan standar pemrograman untuk mengakses aplikasi atau layanan berbasis web. Sebuah perusahaan software atau penyedia layanan berbasis web merilis API mereka kepada publik. Dengannya, pengembang lain dapat mendesain aplikasi yang memanfaatkan layanan mereka.

API memungkinkan sebuah aplikasi berkomunikasi dengan aplikasi lain di Internet melalui serangkaian panggilan (*call*). Sebuah API, berdasarkan definisinya, adalah sesuatu yang mendefinisikan cara dua entitas untuk berkomunikasi. Entitas di sini adalah sebuah software yang nyata berbeda (dalam layanan) dengan software lain.

Dengan API, panggilan-panggilan yang bolak-balik antar aplikasi diatur melalui web service. *Web service* adalah kumpulan standar teknis dan protokol, termasuk XML (*Extensible Markup Language*), bahasa umum yang digunakan oleh aplikasi-aplikasi tersebut selama berkomunikasi di Internet.

API sendiri merupakan sekumpulan kode software yang ditulis sebagai serangkaian pesan XML. Setiap pesan XML berhubungan dengan fungsi spesifik dari aplikasi yang akan diajak berkomunikasi. Sebagai contoh, pada API Facebook, terdapat pesan XML yang berhubungan dengan fungsi spesifik *wall post, wall comment, wall like*.

Pengembang aplikasi pihak ketiga menggunakan pesan-pesan XML yang berhubungan dengan fungsi-fungsi spesifik dari layanan web yang akan diajak berkomunikasi. Pengembang bebas memilih fungsi khusus apa saja yang akan diajak berkomunikasi, dan ditampilkan pada aplikasi rancangannya. Sebagai contoh, kita bisa membuat Facebook client yang hanya menampilkan update status teman-teman kita.

Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan *software*, agar dapat dimanfaatkan oleh pengembang pihak ketiga untuk mendesain aplikasi yang memanfaatkan layanan mereka dengan mudah.

II.2.1 Google Translate API

Google Translate layanan yang disediakan oleh Google Inc. untuk menerjemahkan bagian teks atau halaman web dalam satu bahasa ke bahasa lain.

Layanan Google Translate API diperkenalkan pada 28 April 2006 untuk bahasa Arab. Sebelum Oktober 2007, untuk bahasa selain bahasa Arab, Tionghoa, dan Rusia, Google menggunakan penerjemah berbasis SYSTRAN yang digunakan oleh layanan terjemahan lainnya seperti Babel Fish, AOL, dan Yahoo.

Pada tanggal 26 Mei 2011, Google mengumumkan bahwa Google Translate API akan berhenti berfungsi pada 1 Desember 2011, ‘karena beban ekonomi yang cukup besar disebabkan oleh penyalahgunaan yang luas’.

Pada tanggal 3 Juni 2011, Google mengumumkan bahwa mereka telah membatalkan rencananya untuk menutup Translate API karena adanya tekanan

publik. Dalam pengumuman yang sama, Google mengatakan bahwa mereka akan merilis Translate API versi berbayar.

Dengan adanya Google Translate API ini, maka para pengembang (*developer*) pihak ketiga dapat memanfaatkan Google Translate sebagai penerjemah dalam *website* mereka. Berikut langkah-langkah dalam untuk mengaktifkan Google Translate API :

1. Log in ke ***https://code.google.com/apis/console/?api=translate*** menggunakan akun Google Anda. Jika belum memilikinya, maka silakan registrasi terlebih dahulu.

2. Setelah berhasil login, maka akan muncul tampilan berikut ini jika sebelumnya belum pernah mengaktifkan Google Translate API. Selanjutnya, klik tombol ‘Add project’.



Gambar II.1 Add Project’ untuk mulai menggunakan Google API

3. Lalu muncul beberapa layanan API Google. Dari beberapa layanan API tersebut, cari ‘Translate API’ dan klik tombol ‘Activate’.



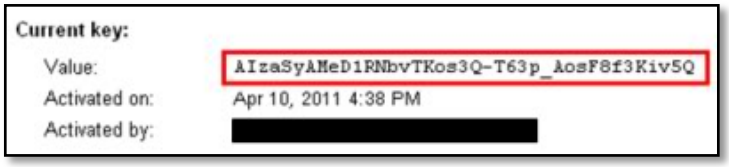
Gambar II.2 Aktifasi Google Translate API

4. Untuk mendapatkan layanan akses ke Google API, setiap *user* akan mendapatkan API Access yang berupa kode tertentu. Kode ini berfungsi sebagai autentikasi kepada setiap user yang mengakses API. Kode API Access Anda dapat dilihat di menu ‘API Access’.



Gambar II.3 Menu API Access untuk mendapatkan kode API Access

5. Setelah Anda klik API Access, maka Anda akan menjumpai suatu kode sebagai berikut (perhatikan kode yang di kotak merah).



Gambar II.4 Key Google Translate API

II.2.2 Konsep Penerjemahan

Google Translate adalah sebuah alat yang secara otomatis menerjemahkan teks dari satu bahasa ke bahasa lain.

Source text adalah teks yang diterjemahkan. ***Source language*** adalah bahasa dimana *source text* ditulis. ***Target language*** adalah bahasa dimana *source text* diterjemahkan.

Operasi-operasi API Penerjemah

Ada tiga metode untuk memanggil Google Translate API :

Tabel II.I Metode Google Translate API

Operasi	**Deskripsi**	***mapping* REST HTTP**
translate	Menerjemahkan *source text* dari *source language* ke *target language*	GET
languages	Mendaftar *source* dan *target languages* yang didukung oleh metode-metode *translate.*	GET
detect	Mendeteksi bahasa dari *source text.*	GET

REST

REST (*Representational State Transfer*) pada Google Translate API agak berbeda dengan REST tradisional. Daripada menyediakan akses untuk sumber-sumber, API menyediakan akses untuk sebuah layanan. Sebagai hasilnya, API menyediakan sebuah *single* URI yang bertindak sebagai layanan *endpoint*.

Translate

Format spesifik untuk *single* Google Translate API URI:

```
https://www.googleapis.com/language/translate/v2?parameters
```

Berikut merupakan contoh bagaimana Google Translate API URI bekerja dalam Translate API.

```
https://www.googleapis.com/language/translate/v2?key=INSERT-YOUR-KEY&q=hallo%20dunia&source=id&target=en
```

Mengetahui bahasa-bahasa yang mendukung

Format spesifik untuk mengembalikan kode-kode daftar bahasa-bahasa adalah sebagai berikut :

```
https://www.googleapis.com/language/translate/v2/languages?{parame
ters}
```

Berikut merupakan contoh bagaimana metode *languages* bekerja dalam Translate API.

```
https://www.googleapis.com/language/translate/v2/languages?key=INS
ERT-YOUR-KEY&target=en
```

Mengetahui source language

Format spesifik untuk mengetahui bahasa dari sebuah teks adalah sebagai berikut :

```
https://www.googleapis.com/language/translate/v2/detect?{parameter
s}
```

Menerjemahkan Text

Kita dapat menerjemahkan teks dari satu bahasa ke bahasa lain dengan mengirimkan sebuah *request* HTTP GET ke URInya. URI untuk mengirimkan *request* memiliki format sebagai berikut :

```
https://www.googleapis.com/language/translate/v2? {parameters}
```

Berikut merupakan contoh yang menspesifikasikan *source language*, menggunakan *source query parameter* :

```
GET https://www.googleapis.com/language/translate/v2?key=INSERT-
YOUR-KEY&source=id&target=en&q=Hallo%20dunia
```

Jika *request* sukses, maka *server* akan memberikan respon dengan kode status HTTP 200 OK dan *properties* data sebagai berikut :

JSON

```
200 OK

{
    "data": {
        "translations": [
            {
```

```
                    "translatedText": "Hello World"
                }
            ]
        }
    }
```

Berikut merupakan contoh lain dengan menggunakan *auto-detected* bahasa dari *source text* :

```
GET https://www.googleapis.com/language/translate/v2?key=INSERT-YOUR-KEY&target=en&q=Hallo%20dunia
```

Jika *request* sukses, maka *server* akan memberikan respon dengan kode status HTTP 200 OK dan *properties* data sebagai berikut :

JSON

```
200 OK

{
    "data": {
        "translations": [
            {
                "translatedText": "Hello World",
                "detectedSourceLanguage": "en"
            }
        ]
    }
}
```

Referensi parameter *query*

Parameter-paramater *query* yang dapat digunakan dengan Translate API dicantumkan dalam tabel berikut.

Tabel II.2 Parameter-paramater *query* pada Google Translate API

Parameter	Arti	Catatan
callback	Fungsi JavaScript yang dijalankan ketika respon diterima.	1. Parameter ini memperbolehkan kita untuk menspesifikasikan fungsi JavaScript untuk

		mengaatur hasil *query* bagi implementasi *client-side* murni. 2. Menyimpan *query* API dalam tag <script>. 3. Menetapkan fungsi *callback* dalam tag <script>.
format	Format	1. Parameter ini memperbolehkan kita untuk menyatakan apakah teks akan diterjemahkan dalam *plain-text* atau HTML. Nilai html menyatakan HTML dan nilai teks menyatakan *plain-text*. 2. Default: format=html.
key	API *key*	API ini memerlukan *key* yang *valid* untuk semua *request.*
prettyprint	Mengembalikan respon dengan bentuk paragraf dan spasi antar baris.	1. Jika prettyprint=true, maka hasil yang dikembalikan oleh server akan dapat dibaca oleh manusia (tercetak dengan baik). 2. Default: prettyprint=true.

source	*Source text*	1. Teks yang akan diterjemahkan. 2. Kita dapat mengulang paramater ini lebih dari satu kali dalam *request* tunggal untuk menerjemahkan teks jamak.
q	*Source language*	1. Bahasa dari sumber teks. Nilai harus merupakan salah satu kode bahasa pada referensi bahasa. 2. Jika bahasa tidak dispesifikasikan, maka sistem akan mencoba untuk mengidentifikasi sumber bahasa secara otomatis.
target	*Target language*	3. Bahasa untuk menerjemahkan sumber teks ke dalam bahasa yang ditargetkan. Nilai harus merupakan salah satu kode bahasa pada referensi bahasa.

Referensi bahasa-bahasa :

Bahasa	Kode Bahasa
Afrika	af
Albania	sq
Arab	ar
Azerbaijani	az
Basque	eu
Bengali	bn
Belarusia	be
Bulgaria	bg
Catalan	ca
Chinese Simplified	zh-CN
Chinese Traditional	zh-TW
Croasia	hr
Czech	cs
Denmark	da
Belanda	nl
Inggris	en
Esperanto	eo
Estonian	et
Filipina	tl
Finnish	fi
Prancis	fr
Galician	gl
Georgian	ka
Jerman	de
Yunani	el
Gujarati	gu

Haitian Creole	ht
Yahudi	iw
Hindi	hi
Hungaria	hu
Islandia	is
Indonesia	id
Irlandia	ga
Itali	it
Jepang	ja
Kanada	kn
Korea	ko
Latin	la
Latvian	lv
Lithuania	lt
Macedonia	mk
Malaysia	ms
Maltese	mt
Norwegia	no
Persia	fa
Polandia	pl
Portugis	pt
Romania	ro
Rusia	ru
Serbia	sr
Slovak	sk
Slovenian	sl
Spanyol	es
Swahili	sw

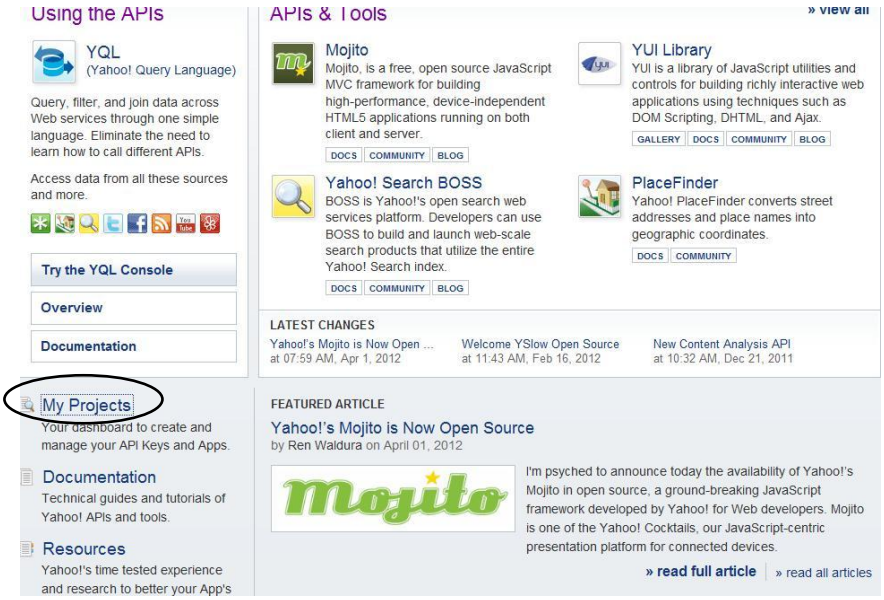
Swedia	sv
Telugu	te
Thailand	th
Turki	tr
Ukrainian	uk
Urdu	ur
Vietnam	vi
Wales	cy
Yiddi	yi
Tamil	ta

II.2.2 Yahoo! Messenger API

Yahoo! Messenger (sering disingkat "Y!M" atau "YM") merupakan program pengirim pesan instan populer yang disediakan oleh Yahoo!. Yahoo! Messenger tersedia secara gratis dan dapat diunduh serta diakses menggunakan Yahoo! ID yang biasa digunakan untuk mengakses layanan Yahoo! yang lainnya, seperti Yahoo! Mail.

Yahoo! Messenger API menawarkan sebuah *platform* terpercaya kepada para pengembang (*developer*) untuk mengelola kontak dan grup daftar, informasi tentang keberadaan, komunikasi instan *real-time* dan transfer data ke dan dari klien. Berikut langkah-langkah dalam mengakses Yahoo! Messenger API *key* :

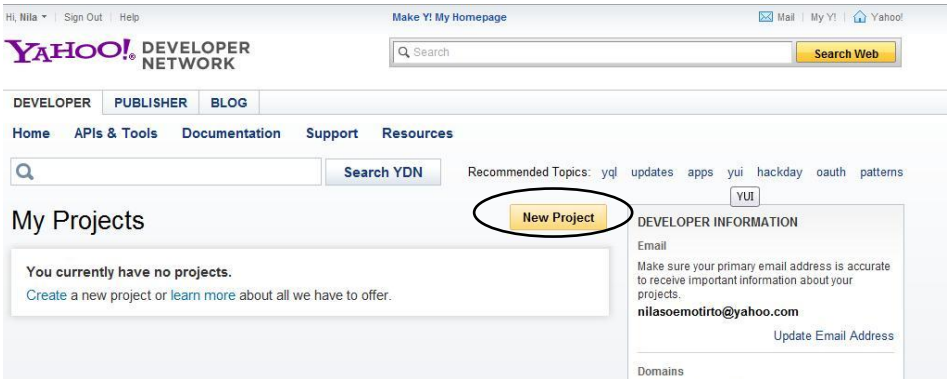
1. Membuat Open Authentication (OAuth) API key menggunakan akun Yahoo dengan mengakses situs ***http://developer.yahoo.com***. Pilih “My Projects”, seperti pada gambar berikut.



Gambar II.5 “My Projects” untuk mulai menggunakan key Yahoo OAuth API

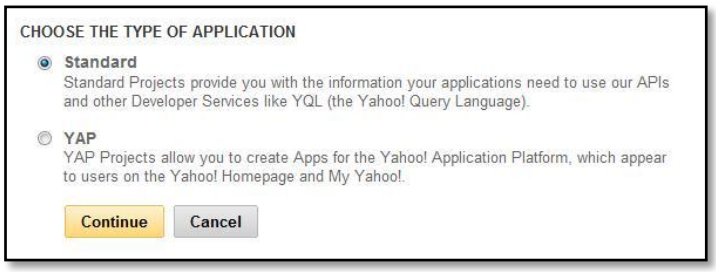
2. Pilih ‘Accept’ pada *Term of Use.*

3. Halaman ini berisi semua proyek aplikasi yang telah Anda buat. Untuk menggunakan Yahoo! Messenger pada aplikasi Anda, maka klik tombol ‘New Project’ seperti pada gambar di bawah ini.



Gambar II.6 “New Project” untuk mulai menggunakan Yahoo Messenger pada aplikasi

4. Kemudian akan muncul kotak dialog sebagai berikut. Pastikan untuk memilih pilihan yang pertama, lalu klik ‘Continue’.



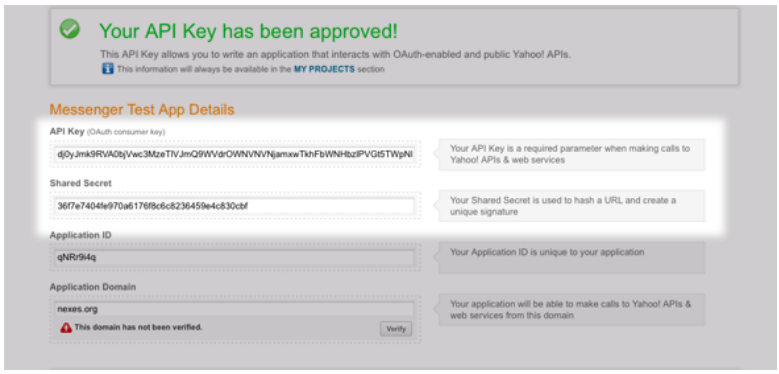
Gambar II.7 Tipe aplikasi yang akan dibangun

5. Pada "Access Scopes", pilih "This app requires access to private user data", seperti pada gambar berikut. Kemudian klik ‘Get API Key’.



Gambar II.8 *Access scopes* pada aplikasi yang akan dibangun

6. Halaman terakhir berikut berisi data OAuth yang terdiri dari dua *field* penting, yaitu : *OAuth Consumer Key* dan *Consumer Secret Key*.



Gambar II.9 *OAuth Consumer Key* dan *Consumer Secret Key*

II.2.2.1 Logging In

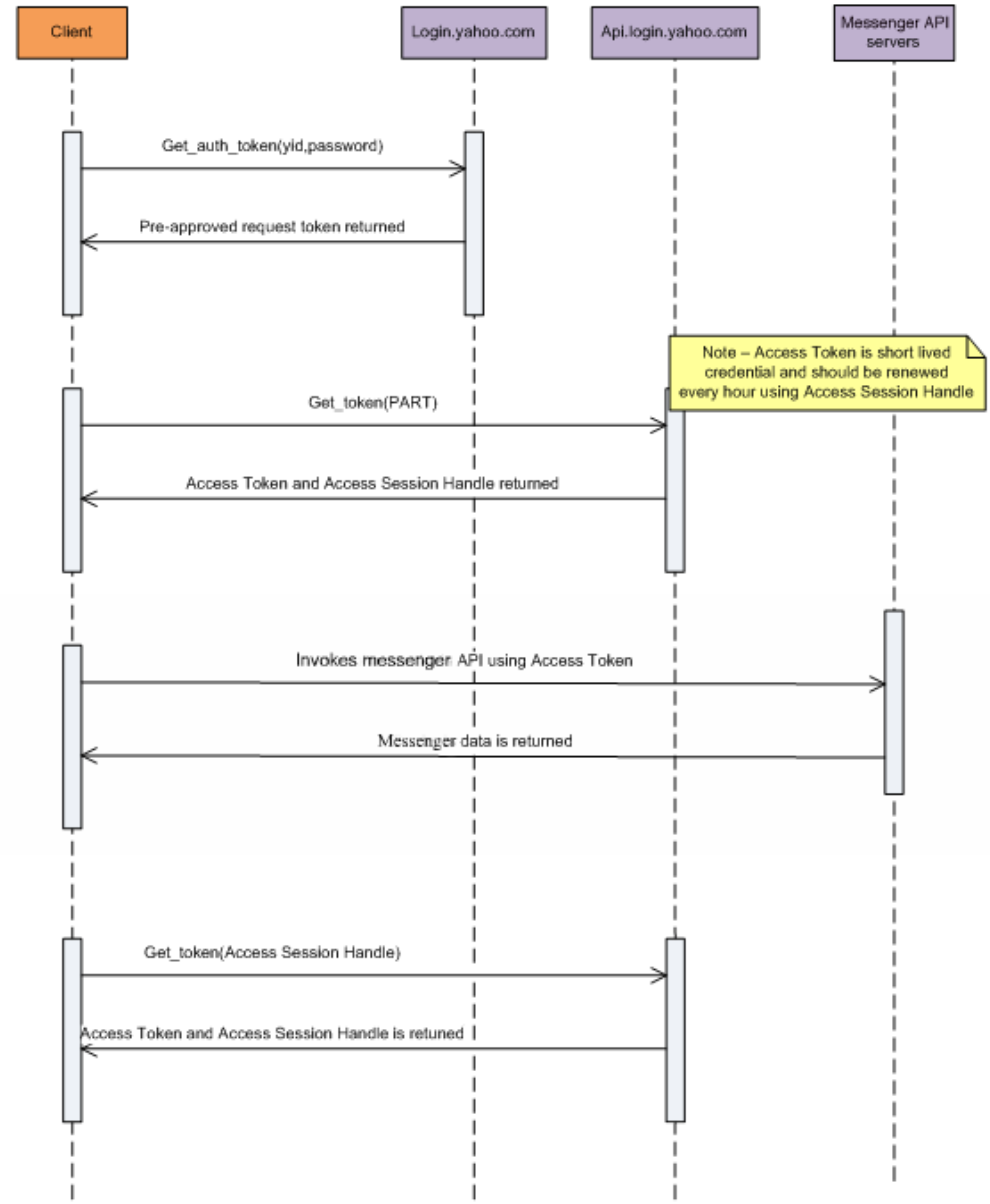
Untuk *login,* Yahoo! Messenger IM SDK memerlukan Yahoo! *Username* dan *password* yang valid. Langkah awal yang harus dilakukan yaitu mendapatkan bukti keaslian 'surat kepercayaan (*credentials)*' dari *server* Yahoo! Login. Hal ini hanya dapat dilakukan dengan menggunakan Yahoo! Direct OAuth API. Terdapat dua API calls yang diperlukan untuk melakukan otentikasi dengan Yahoo! Direct OAuth API. API *call* yang pertama melakukan Pre-Authorized Request Token (PART), dan memerlukan login *username, password,* dan OAuth Consumer Key (API Key) melalui *developer.yahoo.com*. API *call* sebagai berikut :

```
https://login.yahoo.com/WSLogin/V1/get_auth_token?&login=username&
passwd=mypassword&oauth_consumer_key=consumerkey
```

Hasil dari *call* tersebut merupakan sebuah nilai tunggal, `RequestToken`:

```
RequestToken=jUO3Qolu3AYGU1KtB9vUbxlnzfIiFRLP...
```

Token ini nantinya akan digunakan untuk *request* kedua, yang mana akan menukar PART untuk sebuah access token OAuth. Pada Direct OAuth, sebuah aplikasi memasukkan *username*, *password*, dan *consumer key* ke *login server* untuk memuat *Pre-Approved Request Token* (PART). Setelah memuat PART, pemakai menggunakan the *interface* Yahoo Oauth standard untuk memuat Access Token, Access Token Secret, dan *access session handle* (ASH). Berikut *sequence diagram* untuk alur otentikasi :



Gambar II.10 *Sequence diagram* alur otentikasi

II.2.2.2 Kehadiran (*presence*)

Yahoo! Messenger IM SDK *pre-define* empat tingkat kehadiran, diwakili secara internal oleh nilai-nilai bilangan bulat.

Tabel II.3 *Preresence*

Nilai	Tingkatan
-1	*User offline*
0	*User online*
2	*User* sibuk
999	*User idle*

Anda juga dapat memperoleh dan mengatur *Presence Management* API. Me*reset* kehadiran (*presence)* Anda sangat berguna dalam merubah *presence state* atau pesan Anda setelah *login*. Sebagai contoh,*call* ini akan memperbarui *presence state* dan pesan kehadiran untuk menunjukkan bahwa *user* sedang sibuk di sebuah pertemuan.

```
PUT /v1/presence?sid=msgrsessionid
Host: rcore1.messenger.yahooapis.com
Authorization: < Standard OAuth credentials >
Content-Type: application/json;charset=utf-8
Content-Length: 48
{
  "presenceState" : 2,
  "presenceMessage" : "I am in a meeting"
}
```

Apabila tidak terjadi kesalahan, maka respon dari *server* akan berupa sebuah HTTP OK.

```
HTTP/1.1 200 OK
```

II.2.2.3 Mengirimkan Pesan

Message Management API dapat digunakan untuk mengirim pesan ke kontak Yahoo! Messenger yang lainnya. API ini sangat sederhana untuk digunakan, seperti berikut :

```
POST /v1/message/yahoo/targetYahooId?sid=msgrsessionid
Host: rcore1.messenger.yahooapis.com
Authorization: < Standard OAuth credentials >
Content-Type: application/json;charset=utf-8
Content-Length: 25
{
    "message" : "Hey there"
}
```

Apabila tidak terjadi kesalahan, maka respon dari *server* akan berupa sebuah HTTP OK.

```
HTTP/1.1 200 OK
```

II.2.2.4 Logging Out

Berikut merupakan contoh *request log out.*

```
DELETE /v1/session
Host: rcore1.messenger.yahooapis.com
Authorization: < Standard OAuth credentials >
```

Apabila *user* telah *log out,* maka respon dari *server* adalah HTTP OK sederhana.

```
HTTP/1.1 200 OK
```

II.2.2.5 Definisi-definisi Error

Tabel berikut menjelaskan definisi-definisi error untuk Yahoo! Messenger IM SDK.

Tabel II.4 Definisi-definisi error Yahoo! Messenger IM SDK.

Kode-kode Error	Definisi Kode Error
0	OK
1	Internal Server Error
2	User Exists Error
3	User Does Not Exist Error

4	Invalid Credentials Error
6	Max Buddies Error
7	Max Groups Error
8	Group Exists Error
10	Max Ignored Users Error
12	User Is Contact Error
22	Yahoo! Parents Control Error
28	Session Expired Error
30	Ban Error
32	Invalid Group ID Error
33	User in Ignore List Error
42	Max Pending Contact Authorizations Error
-1000	Invalid Sender Error
-1001	Invalid Target Error
-1003	Unsupported Content Type Error
-1004	Unsupported User Agent ID Error
-1005	Invalid Arguments / Bad Message Error
-1006	Underage Error
-1009	Capability Check Error
(default)	Internal Server Error

II.2.3 Google Talk API

Google Talk (GTalk) adalah suatu layanan aplikasi berbasis web dari Google yang dibuat sebagai sarana untuk berkomunikasi melalui pesan instan dan VoIP. Google talk pertama kali dirilis pada tanggal 24 Agustus 2005. Google Talk disediakan secara gratis dan dapat diunduh serta diakses melalui Google. Layanan Google Talk hanya dapat digunakan apabila telah memiliki akun di Google.

Pesan instan yang ada di antara server Google Talk dan kliennya menggunakan protokol terbuka, dan XMPP, yang memungkinkan para pengguna klien XMPP lainnya untuk berkomunikasi dengan para pengguna Google Talk. Sementara itu, VoIP pada google Talk berbasis Jingle protokol. Teknologi yang dipakai dalam jaringan *server* Google tidak dapat diketahui oleh publik. Layanan Google Talk hanya tersedia untuk sistem operasi yang menggunakan Windows 2000, windows XP, windows Server 2003, windows Vista, dan Windows 7. Perangkat lunak Google Talk tersedia juga untuk BlackBerry, iPhone dan T-Mobile G1. Dengan dikeluarkannya gadget pada Google Talk membuat seluruh pengguna yang didukung oleh Adobe Flash Player dapat menggunakan Google Talk.

Dengan adanya Google Talk API, maka kita dapat memanfaatkan layanan-layanan yang ada pada Google Talk seperti menampilkan daftar kontak *user*, melakukan komunikasi *real-time, voice streaming*, *video straming,* dan *file-sharing.*

II.2.3.1 Ekstensi Google Talk XMPP

Pengguna dapat menyesuaikan beberapa pengaturan pada *server* Google Talk untuk disesuaikan dengan pengalaman klien mereka.

Menentukan server yang mendukung

Klien harus belajar terlebih dahulu apakah server mendukung ekstensi ini dengan menggunakan pencarian layanan. Klien harus mengirimkan Bait IQ berikut untuk menemukan fitur apakah yang didukung oleh *server.*

Contoh 1. Permintaan Layanan Klien dalam Pencarian

```
<iq type='get' to='gmail.com'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Apabila *server* mendukung ekstensi setting-setting pengguna, maka respon akan memuat fitur *'google:setting'* seperti berikut ini :

Contoh 2. Respon Layanan Klien dalam Pencarian

```
<iq type='result' to='romeo@gmail.com' from='gmail.com'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='google:setting'/>
    ...
  </query>
</iq>
```

II.2.3.2 Pemberitahuan dari Gmail

Pengantar

Google Talk menggunakan ekstensi XMPP sehingga memungkinkan pengguna untuk melakukan *query* pada akun Gmail untuk *email* dan menerima pemberitahuan ketika ada sebuah pesan baru.

Menentukan server yang mendukung

Klien harus belajar terlebih dahulu apakah server mendukung ekstensi ini dengan menggunakan pencarian layanan. Klien harus mengirimkan Bait IQ berikut untuk menemukan fitur apakah yang didukung oleh *server.*

Contoh 3. Permintaan Layanan Klien dalam Pencarian

```
<iq type='get' to='gmail.com'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Apabila *server* mendukung ekstensi setting-setting pengguna, maka respon akan memuat fitur yang benar seperti di bawah ini :

Contoh 4. Respon Layanan Klien dalam Pencarian

```
<iq type='result' to='romeo@gmail.com' from='gmail.com'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='google:mail:notify'/>
    ...
  </query>
</iq>
```

II.2.3.3 Pemberitahuan Email Baru

Ketika akun pengguna Gmail menerima *email* baru, jika seorang klien sebelumnya telah mengirim *query email* baru kepada *server*, maka *server* akan mengirimkan pemberitahuan kepada klien. Pemberitahuan ini memuat sebuah elemen `<new-mail/>` yang dikualifikasikan oleh 'google:mail:notify' seperti berikut :

Contoh 5. Server memberitahukan email baru

```
<iq type='set'
    from='romeo@gmail.com'
    to='romeo@gmail.com/orchard'
    id='mail-request-2'>
  <new-mail xmlns='google:mail:notify' />
</iq>
```

Contoh 6. Klien mengembalikan nilai sukses

```
<iq type='result'
    from='romeo@gmail.com'
    to='romeo@gmail.com/orchard'
    id='mail-request-2' />
```

II.2.4 Facebook Chat API

Facebook adalah sebuah layanan jejaring sosial dan situs web yang diluncurkan pada Februari 2004 yang dioperasikan dan dimiliki oleh Facebook, Inc. Pada Januari 2011, Facebook memiliki lebih dari 600 juta pengguna aktif. Pengguna dapat membuat profil pribadi, menambahkan pengguna lain sebagai teman dan bertukar pesan, termasuk pemberitahuan otomatis ketika mereka memperbarui profilnya.

Mulai 23 April 2008, Facebook *chat* dirilis ke seluruh pengguna Facebook. Pengguna dapat melakukan *chatting* dengan teman-teman Facebook mereka secara perorangan, atau pengguna dapat chatting dengan beberapa teman-teman secara simultan melalui fitur grup. Beberapa klien *instant messaging* yang baru-baru ini mendukung Facebook diantaranya adalah Yahoo! Messenger versi 11.0 and later, Skype versi 5 or later, AOL Instant Messenger, eBuddy, Flock,

Miranda IM, Trillian, Empathy, Pidgin, Adium, Nimbuzz, FIM (Windows Mobile / Windows Phone 7), Palringo (Windows Mobile), Meebo.

Dengan Facebook Chat API, kita dapat mengintegrasikan Facebook Chat ke produk *instant messaging* baik berbasis web, desktop atau *mobile*. *Instant messaging* terhubung ke Faceook *chat melalui layanan Jabber/XMPP*. Kita dapat mengotentifikasi *user chat* klien dengan menggunakan salah satu dari dua mekanisme berikut : X-FACEBOOK-PLATFORM (Facebook Platform) and DIGEST-MD5 (username/password). Facebook mrekomendasikan untuk menggunakan mekanisme X FACEBOOK PLATFORM untuk terhubung ke Facebook *Chat* karena menyediakan pengalaman pengguna yang lebih baik dengan menggunakan otentikasi Facebook *platform* yang sederhana.

Kami mendukung mekanisme SASL yang disebut X-FACEBOOK-PLATFORM yang memungkinkan klien untuk terhubung ke Facebook *chat* menggunakan otentikasi Facebook. Mekanisme ini lebih dipilih untuk aplikasi apapun yang diorientasikan ke media sosial secara umum atau Facebook secara khusus, terutama aplikasi yang sudah terintegrasi dengan Facebook Platform.

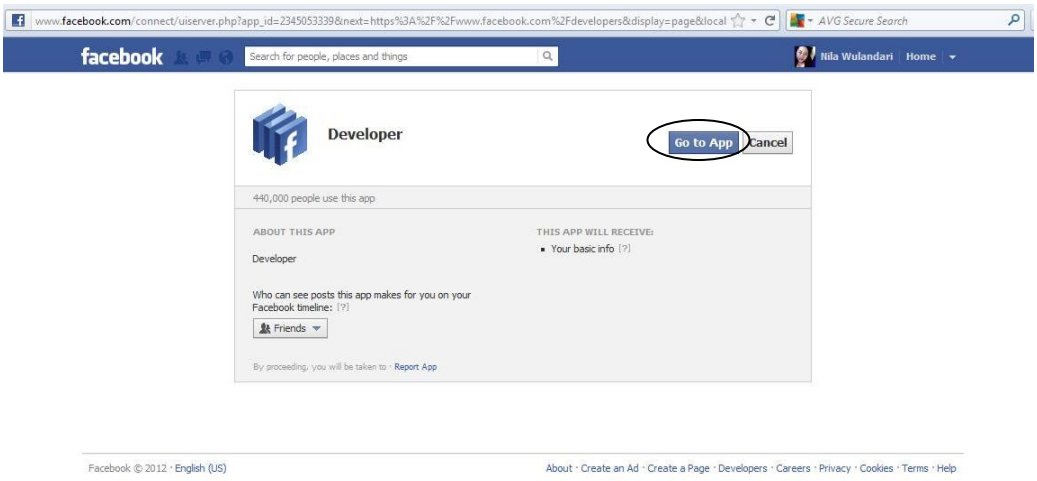
Mekanisme SASL DIGEST-MD5 tersedia untuk mendukung XMPP tradisional atau multi-protokol klien IM yang tidak disesuaikan untuk (atau bahkan sadar) Facebook. Mekanisme ini memerlukan mendorong pengguna untuk sandi; oleh karena itu, itu hanya boleh digunakan bila diperlukan.

Mekanisme SASL DIGEST-MD5 tersedia untuk mendukung XMPP tradisional atau multi-protokol klien IM yang tidak disesuaikan untuk Facebook. Mekanisme ini memerlukan sandi pengguna, oleh karena itu, itu hanya boleh digunakan bila diperlukan.

Facebook saat ini dapat diintegrasikan dengan aplikasi lain, sebagai jembatannya Anda harus memiliki API Key Facebook. Berikut langkah-langkah untuk mendapatkan API Key Facebook :

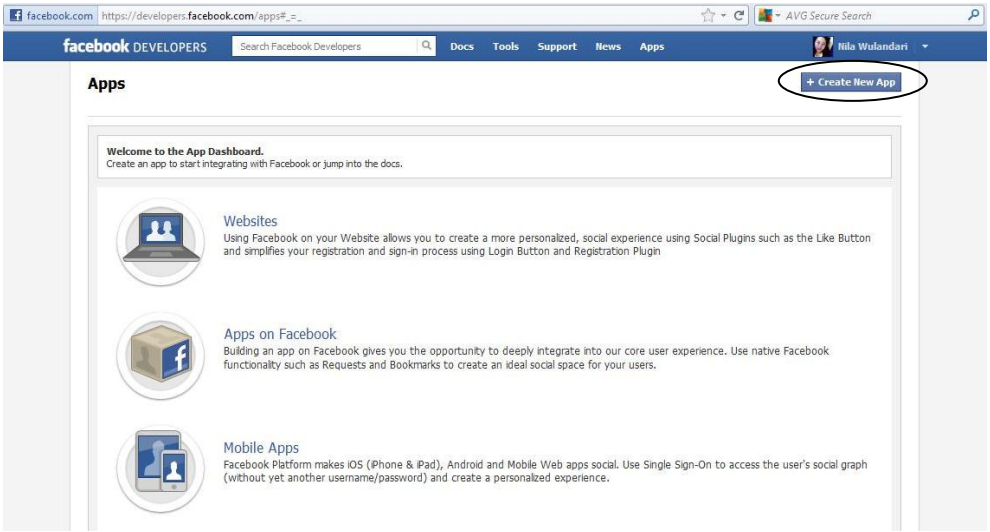
1. Log in ke ***http://www.facebook.com/developers/createapp.php*** menggunakan akun Facebook Anda. Jika belum memilikinya, maka silakan registrasi terlebih dahulu.

2. Kemudian akan muncul tampilan berikut. Klik “Go to App”.



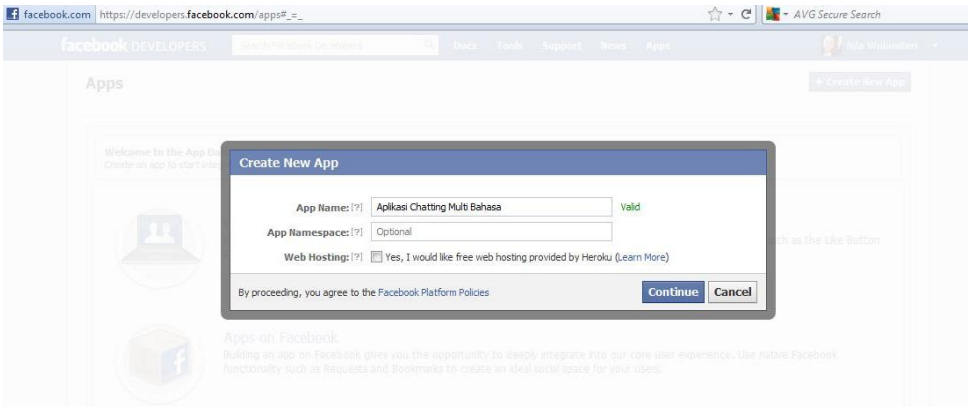
Gambar II.11 Akses untuk masuk ke halaman “Facebook Developers”

3. Kemudian akan muncul tampilan berikut. Klik “Create New App”.



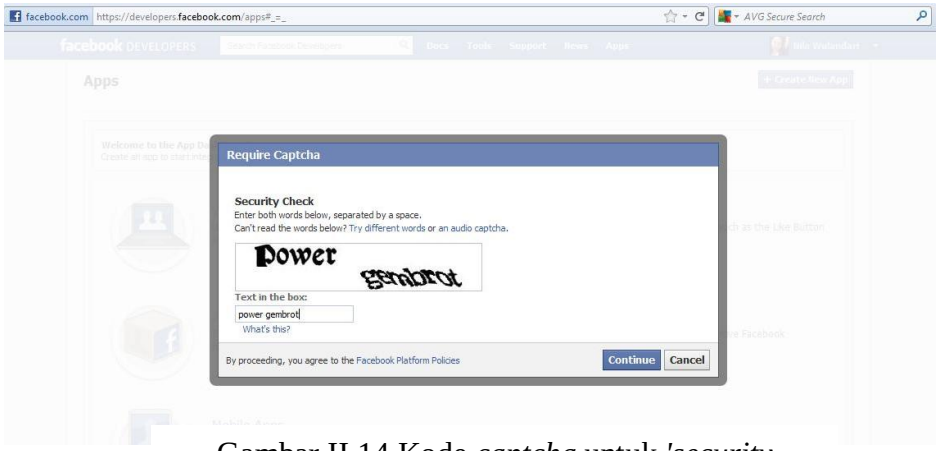
Gambar II.12 Halaman awal “Facebook Developers”

4. Pada halaman berikut masukkan 'nama aplikasi' yang akan Anda buat pada kolom App Display Name lalu klik “Continue”.



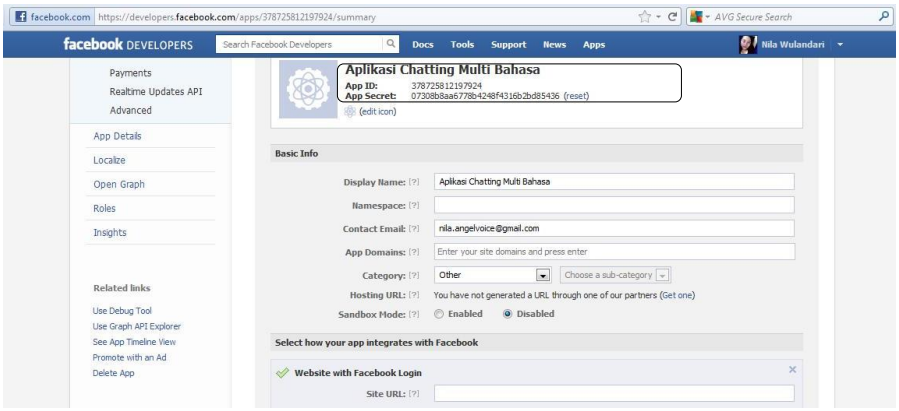
Gambar II.13 *Form* pengisian nama aplikasi

5. Pada halaman berikut masukkan kode *captcha* sesuai dengan huruf-huruf yang ditampilkan kemudian klik “Continue”



Gambar II.14 Kode *captcha* untuk *'security*

6. Halaman terakhir berikut berisi App ID dan App Secret



Gambar II.15 App ID dan App Secret

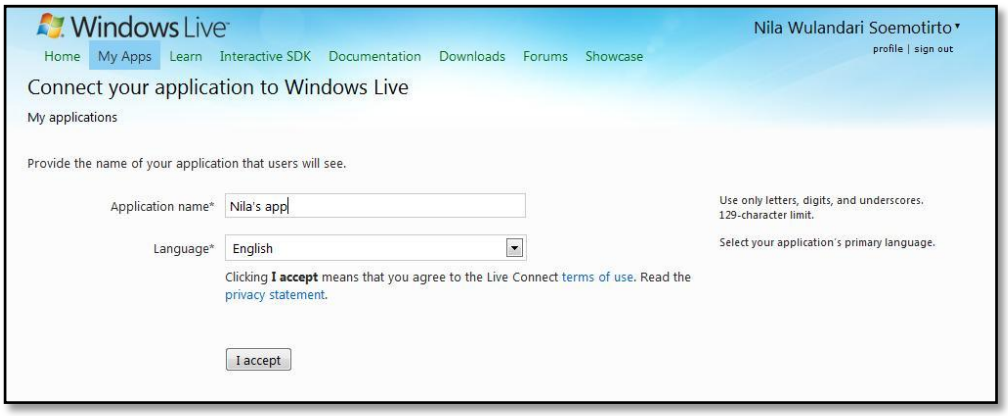
II.2.5 MSN / Windows Live Messenger API

Windows Live Messenger merupakan salah program pengirim pesan instan yang awalnya bernama MSN Messenger lalu Windows Messenger. Program ini diluncurkan oleh Microsoft sebagai pengganti MSN Messenger pada tanggal 19 Juni 2006.

Dengan Windows Live Messenger API, kita dapat mengelola daftar kontak teman pada akun Windows Live Messenger, melakukan *chatting,* dan fitur-fitur lain pada Windows Live Messenger. Berikut langkah-langkah dalam mengakses Windows Live Messenger API :

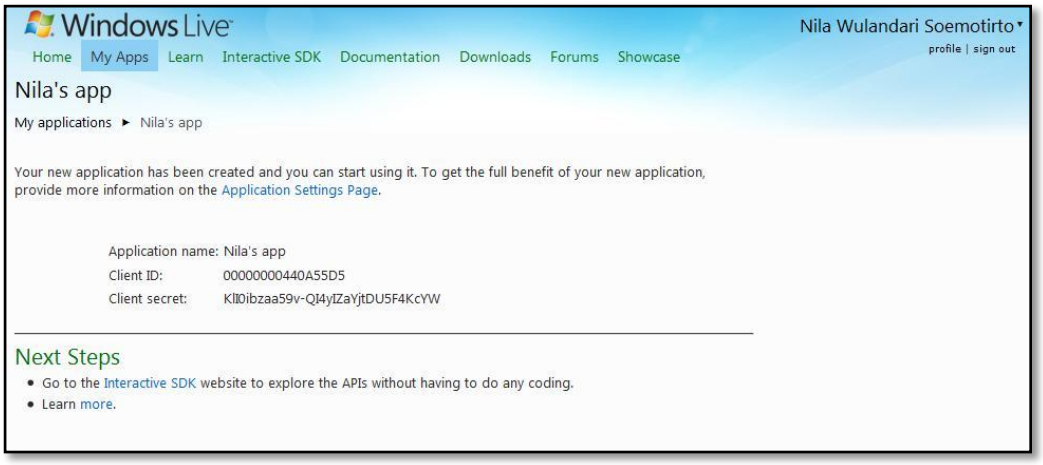
1. Log in menggunakan akun Windows Live Messenger atau sign up bila belum memiliki akun. Kemudian akses situs ***https://manage.dev.live.com/*** dan klik “My Apps”.

2. Kemudian lengkapi *form* aplikasi baru "*Connect your application to Windows Live*". Masukkan *application name*, pilih *language* dan klik "I accept ".



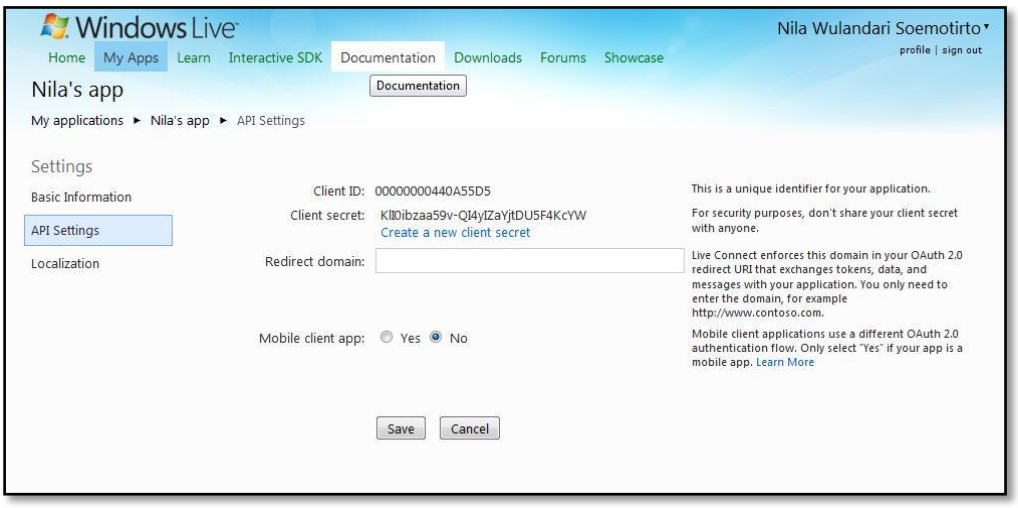
Gambar II.16 *Form* untuk aplikasi baru

3. Akan muncul halaman berikut yang berisi Client ID dan Client Secret.



Gambar II.17 Client ID dan Client Secret

4. Berikut merupakan halaman yang berisi API Settings



Gambar II.18 *API Settings*

II.2.5.1 Mulai Menggunakan Messenger dengan XMPP

Mendapatkan ID Klien

Sebelum aplikasi Anda dapat melakukan *sign in* ke layanan Messenger XMPP, Anda memerlukan ID klien. ID klien adalah kode unik yang mengidentifikasi aplikasi Anda untuk layanan otentikasi Live Connect dan ditukar dengan akses token.

Mendapatkan akses token

Setelah Anda memiliki ID klien (tergantung pada jenis app Anda), aplikasi Anda perlu memberi info ini untuk layanan otentikasi Live Connect untuk ditukar akses token. Token akses ini memungkinkan aplikasi Anda untuk *sign in* ke layanan Messenger XMPP untuk pengguna tertentu yang telah setuju.

Sign in

Setelah Anda memiliki akses token, aplikasi Anda dapat sign in ke layanan Messenger XMPP atas nama pengguna.

Layanan Messenger XMPP menggunakan otentikasi SASL X-MESSENGER-OAUTH2. App Anda mengirim token akses sebagai tanda SASL

untuk otentikasi, seperti yang ditunjukkan dalam pertukaran parsial dari protokol info berikut, dimana ACCESS_TOKEN mewakili akses token Anda.

```
...
< <stream:features
xmlns:stream="http://etherx.jabber.org/streams"><mechanisms
xmlns="urn:ietf:params:xml:ns:xmpp-sasl"><mechanism>X-
MESSENGER-OAUTH2 </mechanism></mechanisms></stream:features>
> <auth xmlns="urn:ietf:params:xml:ns:xmpp-sasl"
mechanism="X-MESSENGER-OAUTH2">ACCESS_TOKEN</auth>
< <success xmlns="urn:ietf:params:xml:ns:xmpp-sasl" />
...
```

Menambah teman pada Messenger

Untuk menambah teman baru *messenger,* aplikasi Anda harus menggunakan Jabber ID (JID) milik teman Anda dan bukan dengan menggunakan ID akun Microsoft milik teman Anda. Apabila dalam menambah teman aplikasi Anda tidak menggunakan JID milik teman Anda, maka aplikasi Anda tidak akan bias menampilkan info kehadiran teman Anda.

Berikut adalah contoh penggunaan ID akun Microsoft ke layanan dan mengambil JID yang sesuai :

```
< <iq from='john@contoso.com' to='mary@contoso.com'
type='get' id='id1'>
    <getjid
xmlns='http://messenger.live.com/xmpp/jidlookup'></getjid>
</iq>

> <iq from='john@contoso.com' to='mary@contoso.com'
type='result' id='id1'>
    <getjid
xmlns='http://messenger.live.com/xmpp/jidlookup'>
        <jid>12345@messenger.live.com</jid>
    </getjid>
</iq>
```

II.3 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (*Object Oriented Programming*/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Pemrograman beorientasi objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada

permulaan tahun 1970. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing *object* harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan *object* yang lain.

Konsep dasar dari pemrograman berorientasi objek menekankan konsep berikut:

1. *Class*

Class merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-*programmer* sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. *Object*

Object membungkus data dan fungsi menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program berorientasi objek.

3. Abstraksi

Abstraksi merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. Enkapsulasi

Enkapsulasi merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut.

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak. Hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. Polimorfisme

Poliformisme merupakan konsep yang memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu object agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda. Model data berorientasi objek disebut dapat memberi fleksibilitas yang lebih, kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar.

Berikut bahasa pemrograman yang mendukung pemrograman berorientasi objek :

1. Visual Foxpro
2. Java
3. C++
4. Pascal (bahasa pemrograman)
5. Visual Basic.NET
6. SIMULA
7. Smalltalk
8. Ruby
9. Python
10. PHP
11. C#
12. Delphi
13. Eiffel
14. Perl

15. Adobe Flash AS 3.0

Dengan menggunakan pemrograman berorientasi objek, maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

Pemrograman berorientasi objek memliki beberapa keuntungan seperti :

1. Program lebih mudah dibaca dan dipahami, dan pemrograman berorientasi obyek mengontrol kerumitan program hanya dengan mengijinkan rincian yang dibutuhkan untuk programmer.

2. Dapat menggunakan objek-objek sesering yang diinginkan, kita dapat menyimpan objek-objek yang yang dirancang dengan baik ke dalam sebuah tolkit rutin yang bermanfaat yang dapat disisipkan kedalam kode yang baru dengan sedikit perubahan atau tanpa perubahan pada kode tersebut.

II.3.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

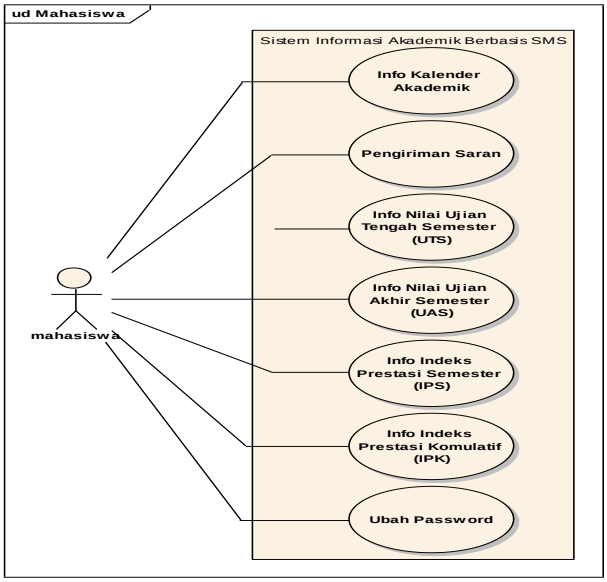
UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem. UML menyediakan beberapa macam diagram untuk memodelkan aplikasi berorientasi objek, diantaranya :

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan proses bisnis berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan aktor. Aktor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh aktor. Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Aktor yang melakukan operasi dihubungkan dengan garis lurus ke use case.

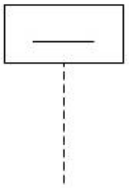
Contoh use case diagram :



Gambar II.19 Contoh Use Case Diagram

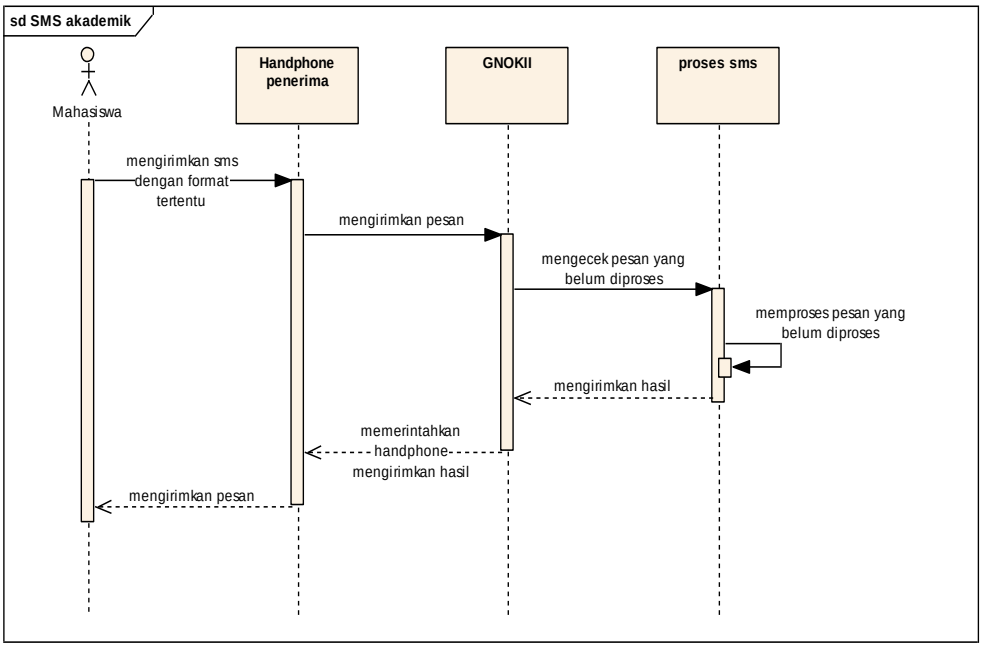
2. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan output tertentu.

Simbol	Keterangan
	Object Timeline
	Aktivasi
	Message

Gambar II.20 Simbol dan keterangan pada sequence diagram

Contoh Sequence Diagram :



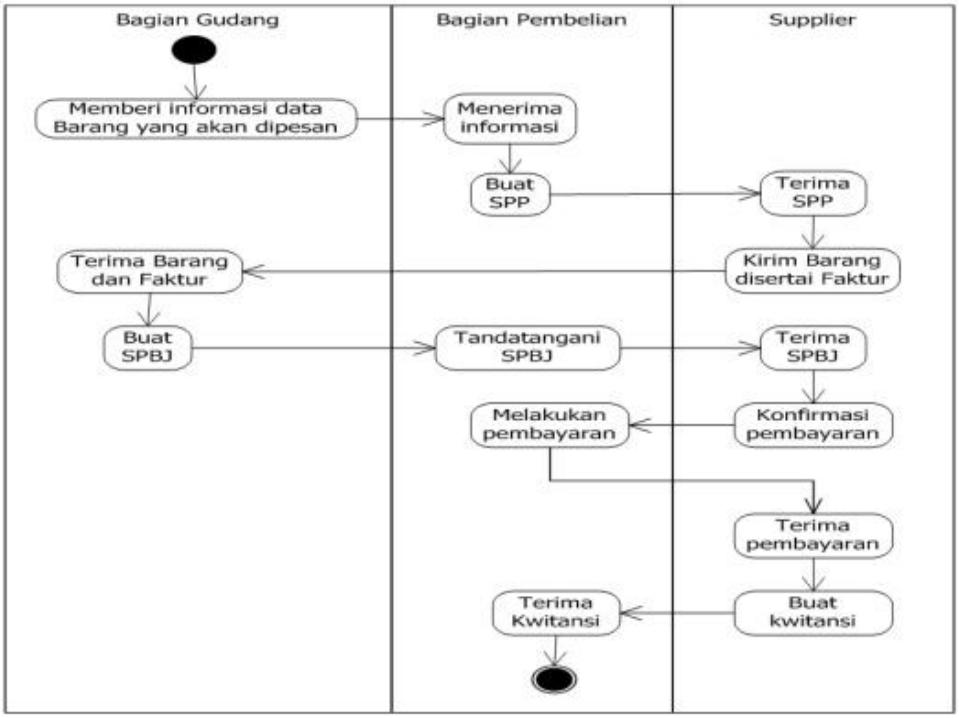
Gambar II.21 Contoh Sequence Diagram

3. Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, kemungkinan yang mungkin terjadi, dan bagaimana merekaberakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Contoh activity diagram :



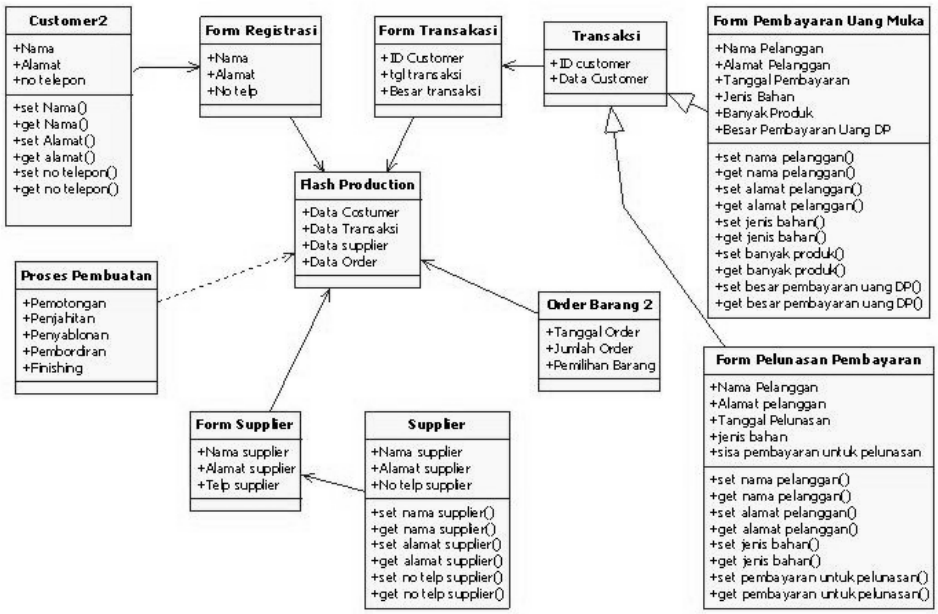
Gambar II.22 Contoh Activity Diagram

4. Class Diagram

Class diagram mendeskripsikan jenis – jenis obyek dalam sistem dan berbagai macam hubungan statis yang terjadi1. Class diagram juga menunjukkan property dan operasi

sebuah Class dan batasan yang terdapat dalam hubungan dengan obyek. Class diagram merupakan alat terbaik dalam perancangan perangkat lunak. Class diagram membantu pengembang mendapatkan struktur sistem dan menghasilkan rancangan sistem yang baik.

Contoh class diagram :



Gambar II.23 Contoh Class Diagram

II.4 Teknologi Web

II.4.1 Internet

Internet adalah rangkaian atau jaringan sejumlah komputer yang saling berhubungan. Internet berasal dari kata interconnected-networking. Internet merupakan jaringan global yang menghubungkan suatu jaringan (network) dengan jaringan lainnya di seluruh dunia. Media yang menghubungkan bisa berupa kabel, kanal satelit maupun frekuensi radio.

Jaringan internet bekerja bekerja berdasarkan suatu protokol (aturan). TCP/IP yaitu Transmission Control Protocol / Internet Protocol. TCP/IP adalah protokol standar yang digunakan untuk menghubungkan jaringan-jaringan di dalam internet sehingga data dapat dikirim dari satu komputer ke komputer lainnya. Setiap komputer diberikan suatu nomor unik yang disebut dengan alamat IP.

Sejarah internet dimulai dari ARPANet, yaitu sebuah proyek Departemen Pertahanan Amerika Serikat. Pada tahun 1969 dilakukan sebuah riset yaitu

bagaimana cara menghubungkan suatu komputer dengan komputer lainnya atau membentuk suatu jaringan. Di tahun 1970 mereka berhasil menghubungkan lebih dari 10 komputer yang membentuk jaringan. Kemudian tahun 1973 jaringan ARPANet mulai dikembangkan di luar Amerika Serikat. Sejarah internet berlanjut ketika komputer di University College di London ikut bergabung dengan jaringan ARPANet. Di tahun-tahun selanjutnya jaringan ini semakin berkembang.

Pada tahun 1982, karena banyaknya komputer yang bergabung maka dibutuhkan sebuah protokol resmi yang menghubungkan semua komputer dan jaringan dan dibentuklah TCP/IP. Di tahun 1984 kemudian diperkenalkan sistem penamaan domain yang dikenal dengan DNS (Domain Name System). Internet kemudian tumbuh secara pesat di tahun 1990-an. Demikianlah sejarah internet secara singkat.

Beberapa hal yang berhubungan dengan internet :

1. WWW (World Wide Web)

WWW (World Wide Web) merupakan bagian dari internet yang cepat berkembang dan paling populer. WWW merupakan jaringan beribu-ribu komputer yang dikategorikan menjadi dua yaitu : client dan server. Web terdiri dari dua komponen dasar yaitu:

a. Web Server

Sebuah komputer (server) dan *software* yang menyimpan dan mendistribusikan data komputer lainnya melalui jaringan internet.

b. Web Browser

Adalah *software* yang dijalankan pada komputer pemakai (client) yang meminta informasi dari server web dan menampilkannya sesuai dengan file data itu sendiri.

2. HTTP (Hypertext Transfer Protocol)

HTTP (Hypertext Transfer Protocol) adalah protokol yang menentukan aturan yang perlu diikuti oleh web browser dan web server. HTTP merupakan protokol standar yang digunakan dalam mengakses dokumen HTML.

HTTP juga mengatur aksi-aksi apa saja yang harus dilakukan oleh web server dan juga web browser sebagai respon atas perintah-perintah yang ada pada

protokol HTTP ini. Sebagai contoh, ketika kita mengetikkan suatu alamat atau URL pada internet browser, maka sebenarnya web browser akan mengirimkan perintah HTTP ke web server.

Web server kemudian akan menerima perintah ini dan melakukan aktivitas sesuai dengan perintah yang diminta oleh web browser (misalnya akses ke database, file, e-mail dan lain sebagainya). Hasil aktivitas tadi akan dikirimkan kembali ke web browser untuk ditampilkan kepada pengguna.

Protokol HTTP pertama kali dipergunakan dalam WWW pada tahun 1990. Pada saat tersebut yang dipakai adalah protokol HTTP versi 0.9. Versi 0.9 ini adalah protokol transfer dokumen secara mentah, maksudnya adalah data dokumen dikirim sesuai dengan isi dari dokumen tersebut tanpa memandang tipe dari dokumen.

Kemudian pada tahun 1996 protokol HTTP diperbaiki menjadi HTTP versi 1.0. Perubahan ini untuk mengakomodasi tipe-tipe dokumen yang hendak dikirim beserta enkoding yang dipergunakan dalam pengiriman data dokumen.Sesuai dengan perkembangan infrastruktur internet maka pada tahun 1999 dikeluarkan HTTP versi 1.1 untuk mengakomodasi proxy, cache dan koneksi yang persisten.

3. URL (Uniform Resource Locator)

URL adalah suatu sarana untuk menentukan informasi pada suatu web server. URL dapat diibaratkan sebagai suatu alamat. URL terdiri dari :

1. Protokol yang digunakan oleh suatu browser untuk mengambil informasi.

2. Nama komputer (server) dimana informasi tersebut berada.

3. Jalur atau path serta nama file dari suatu informasi.

URL pertama kali diciptakan oleh Tim Berners-Lee pada tahun 1991 agar penulis-penulis dokumen dokumen dapat mereferensikan pranala ke World Wide Web. Sejak 1994, konsep URL telah dikembangkan menjadi istilah Uniform Resource Identifier (URI) yang lebih umum sifatnya.

4. File Transfer Protocol (FTP)

FTP adalah suatu protokol yang berfungsi untuk tukar-menukar file dalam suatu *network*.

Dua hal yang penting dalam FTP yaitu :

1. FTP *server* adalah suatu server yang menjalankan software yang berfungsi untuk memberikan layanan tukar menukar file dimana server tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan (request) dari FTP client.
2. FTP *client* adalah komputer yang merequest koneksi ke FTP server untuk tujuan tukar menukar file. Setelah terhubung dengan FTP server, maka client dapat men-download, meng-upload, merename, men-delete, dan lain-lain sesuai dengan *permission* yang diberikan oleh FTP server.

Tujuan dari FTP server adalah sebagai berikut :

1. Untuk tujuan *sharing* data.
2. Untuk menyediakan tempat penyimpanan bagi user.
3. Untuk menyediakan transfer data yang reliable dan efisien.

5. Domain

Domain merupakan nama unik yang diberikan untuk mengidentifikasi nama server komputer seperti web server atau email server di jaringan komputer ataupun internet. Nama domain berfungsi untuk mempermudah pengguna di internet pada saat melakukan akses ke server, selain juga dipakai untuk mengingat nama server yang dikunjungi tanpa harus mengenal deretan angka yang rumit yang dikenal sebagai IP address. Nama domain ini juga dikenal sebagai sebuah kesatuan dari sebuah situs web seperti contohnya “wikipedia.org”. Nama domain kadang-kadang disebut pula dengan istilah URL, atau alamat website.

6. Hosting

Hosting biasa disebut dengan Web-Hosting, adalah perusahaan yang menyediakan layanan dan penyimpanan Web bagi perorangan atau perusahaan. Dengan kata lain Hosting adalah salah satu bentuk layanan penyewaan ruang simpan data (space) yang digunakan untuk menyimpan data – data website agar halaman website tersebut bisa diakses dari mana saja. Data web tersebut meliputi file – file html, php script, cgi script, css, image, database, dan file lain yang dibutuhkan untuk menampilkan halaman web.

Ada beberapa jenis layanan hosting yaitu :

1. Shared Hosting adalah menggunakan server hosting bersama sama dengan pengguna lain satu server dipergunakan oleh lebih dari satu nama domain. Artinya dalam satu server tersebut terdapat beberapa account yang dibedakan antara account satu dan lainnya dengan username dan password.
2. VPS, Virtual Private Server, atau juga dikenal sebagai Virtual Dedicated Server merupakan proses virtualisasi dari lingkungan software sistem operasi yang dipergunakan oleh server. Karena lingkungan ini merupakan lingkungan virtual, hal tersebut memungkinkan untuk menginstall sistem operasi yang dapat berjalan diatas sistem operasi lain.
3. Dedicated Server adalah penggunaan server yang dikhususkan untuk aplikasi yang lebih besar dan tidak bisa dioperasikan dalam shared hosting atau virtual dedicated server. Dalam hal ini, penyediaan server ditanggung oleh perusahaan hosting yang biasanya bekerja sama dengan vendor.
4. Colocation Server adalah layanan penyewaan tempat untuk meletakkan server yang dipergunakan untuk hosting. Server disediakan oleh pelanggan yang biasanya bekerja sama dengan vendor.

II.5 PHP

PHP singkatan dari Personal Home Page Tools, adalah sebuah bahas script berjenis server side yang menyatu dengan HTML. Sintaks dan perintah-perintah yang dimasukkan akan sepenuhnya dijalankan dan dikerjakan oleh server dan disertai pada halaman HTML biasa. PHP bertujuan untuk membuat aplikasi-aplikasi yang dijalankan diatas taknologi Web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada Web browser, tetapi prosesnya secara keseluruhan dijalankan dan dikerjakan di Web server.

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang PHP: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

II.5.1 Sintaks Dasar PHP

Ada empat macam cara penulisan kode PHP, yaitu :

1. <? echo (“ini adalah script PHP\n”); ?>
2. <? php echo(“ini adalah script PHP\n”); ?>
3. <script language=”php”>

echo (“tulis pakai ini jika HTML editor yang dipakai tidak mengenali PHP”);

</script>

Kita dapat memilih salah satu dari tiga cara tersebut. Tapi yang paling sering digunakan adalah cara pertama dan kedua. Perhatikan bahwa tiap akhir baris harus selalu diberi tanda titik koma (;).

Seperti pada bahasa pemograman lain, kita pun bisa meletakkan baris komentar pada program. Pada PHP caranya adalah dengan meletakkan komentar tersebut disebelah kanan tanda // jika komentar satu baris /* dan /* jika komentar lebih dari satu baris.

II.5.2 Tipe Data pada PHP

PHP mengenal beberapa tipe data berikut:

1. Integer

Yang termasuk dalam tipe data ini adalah bilangan bulat (tidak pakai koma).

2. Floating Point Number

Disebut juga bilangan pecahan. Terdapat tanda titik yang merupakan pemisah antara bagian bulat dan pecahaan.

3. String

String merupakan tipe data yang terdiri dari huruf (teks), angka dan karakterkarakter lainnya.

Teks, angka dan karakter lainnya dapat digabungkan jika ingin membentuk sebuah kalimat atau pernyataan tertentu.

II.5.3 Kelebihan-kelebihan yang Dimiliki PHP

1. PHP termasuk server-side programming, artinya skrip kode diakses di sisi server, dan browser hanya menerima jadi berupa kode HTML. Dengan begitu keamanan kode PHP terjaga.
2. PHP mudah dibuat dan memiliki kecepatan akses yang tinggi.
3. PHP termasuk bahasa embedded, artinya bisa ditempel atau diletakkan dalam tag HTML.
4. PHP dapat dijalankan diberbagai Platform, selain dijalankan di sistem operasi keluarga. *NIX (Linux, UNIX, BSD, dsb), PHP juga bisa berjalan di Windows 9x/ME/XP/NT, dan Machintosh.
5. PHP dapat dijalankan di berbagai Web Server. Bukan hanya Apache, PHP bisa dijalankan di Web Server IIS, PWS, Xitami dan lain sebagainya.
6. PHP mendukung berbagai database, diantaranya : Oracle, Sysbase, mSQL, MySQL, Solid, Generic ODBC, Postgres SQL.

7. PHP mendukung komunikasi dengan layanan lain melalui protokol IMAP, SNMP, NNTP, POP3, dan HTTP.

II.6 HTML

HTML (Hyper Text Markup Language) adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser web* seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan *browser*.

Tag adalah teks khusus (markup) yang terdiri dari nama tag yang diapit oleh karakter "<" (tanda lebih kecil) dan ">" (tanda lebih besar). Contoh tag <body> : adalah tag dengan nama tag body. Tag umumnya ditulis secara pasangan yaitu tag pembuka dan tag penutup. Contoh tag pembuka : <body> dan tag penutup </body> (ditambah tanda "/").

Contoh HTML sederhana :

```
<html>
<head>
<title>Judul Halaman</title>
</head>
<body>
<h1>Heading</h1>
<p>Isi paragraf disini.</p>
```

```
</body>
</html>
```

II.6.1 Instruksi-instruksi pada HTML

Dalam HTML terdapat bermacam-macam instruksi. Disini akan dijelaskan beberapa instruksi HTML yang akan digunakan dalam pembuatan website. Instruksi tersebut adalah :

1. Sepasang container tag <HEAD> dan </HEAD> untuk membuat bagian head dari dokumen. Di dalam container tag ini digunakan juga untuk menulis judul dokumen dengan perintah <TITLE> dan </TITLE>. Untuk membuat tubuh dari dokumen digunakan <BODY> dan diakhiri </BODY>.
2. Untuk memformat teks dapat digunakan perintah-perintah sebagai berikut :
   a. <H1> digunakan untuk menuliskan heading.HTML mendukung enam tingkat heading.

   b. <P> digunakan untuk memberikan jarak antara paragraf dan <BR> digunakan untuk membuat baris baru.

   c. <I>, <B>, dan <U> digunakan untuk membuat teks menjadi miring, tebal, dan bergaris bawah.

   d. <BASEFONT>, <FONT>, dan <CENTER> dapat digunakan untuk memanipulasi ukuran font serta merata tengahkan teks.

   e. <IMG SRC= "/URL/nama_file"> digunakan untuk menempatkan gambar dalam dokumen HTML.

   f. <A HREF= nama file, nama URL> diakhiri dengan </A> digunakan untuk membuat sebuah hypertext link dalam dokumen HTML. Link tersebut dapat berupa file images, teks, dokumen HTNL, URL, ataupun link ke e-mail.

   g. <BODY BGCOLOR="kode warna"TEXT="kode warna" LINK="kode warna" VLINK="kode warna"> untuk memberi warna pada latarbelakang, teks, link yang telah maupun belum dikunjungi pada dokumen HTML.

Secara garis besar, terdapat 4 jenis elemen dari HTML:

1. Structural.

   Tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, <h1>Teknik</h1> akan memerintahkan browser untuk menampilkan "Teknik" sebagai teks tebal besar yang menunjukkan sebagai Heading 1

2. Presentational.

   Tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut , contohnya boldface akan menampilkan bold (cetak tebal). Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks.

3. Hypertext.

   Tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain, contoh Universitas Komputer Indonesia akan menampilkan Universitas Komputer Indonesia sebagai sebuah *hyperlink* ke URL tertentu

4. Elemen widget yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>).

II.7 Javascript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa yang sekarang disebut JavaScript ini dulunya dinamai "LiveScript" yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2 yang sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman "Java") pada masa itu, maka Netscape memberikan nama "JavaScript" kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai "Jscript" di browser milik mereka yaitu Internet Explorer 3. JavaScript sendiri merupakan modifikasi dari bahasa pemrograman C++ dengan pola penulisan yang lebih sederhana dari bahasa pemrograman C++.

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan di sisi klien. Jika kita berbicara dalam konteks web, sederhananya kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. JavaScript adalah bahasa script (bahasa pemrograman yang dapat memegang kontrolaplikasi) yang berbasis pada bahasa pemrograman Java. Cara kerja JavaScript adalah mengakses elemen pada HTML dan membuat aksi jika elemen-elemen HTML itu mengalami perubahan, misalnya berubahnya warna halaman web begitu sebuah tombol diklik. JavaScript membuat sebuah halaman web menjadi lebih dinamis. JavaScript yang digunakan pada halaman web merupakan *client side scripting* yang berarti bahwa web browser mengidentifikasi dan menjalankan skrip program yang disisipkan dalam dokumen web (yang diterima dari server), dan mungkin memperbarui tampilan halaman di komputer pemakai (*user*) tanpa mengirimkan permintaan (*request*) baru kepada web server. Karena JavaScript pada halaman web adalah client side scripting, maka kode JavaScript yang ditulis satu file dengan dokumen HTLM akan ditampilkan ketika sebuah halaman web dilihat *source code*-nya.

Secara fungsional, JavaScript digunakan untuk menyediakan akses skrip pada objek yang dibenamkan (*embedded*) . Contoh sederhana dari penggunaan JavaScript adalah membuka halaman pop up , fungsi validasi pada form sebelum data dikirimkan ke server, merubah gambar kursor ketika melewati objek tertentu, dan lain-lain. Yang harus diperhatikan dalam pengelolaan pemrograman JavaScript diantaranya JavaScript adalah bahasa pemrograman yang case sensitive , yang artinya JavaScript membedakan huruf kecil dan huruf besar. Hal ini sama seperti bahasa pemrograman Turbo C atau C++ dimana huruf "A" tidak sama dengan huruf "a".

JavaScript dapat ditulis secara inline atau satu file dengan dokumen HTML atau juga dapat ditulis pada file terpisah dengan ekstensi .js. JavaScript yang ditulis satu file dengan dokumen HTML dapat ditulis sebagai berikut :

```
<html>

<head>.....</head>
```

```
<script type=”text/javascript” language=”JavaScript”>
/* kode JavaScript dapat ditulis disini */
</script><body>...........</body></html>
```

atau dapat juga ditulis di tengah tag body sebagai berikut :

```
<html>
<head></head><body>...........
<script type=”text/javascript” language=”JavaScript”>
</script>
</body>
</html>
```

II.7.1 Kelebihan dan Kekurangan Javascript

Kelebihan dari bahasa pemrograman Javascript adalah sebagai berikut :

1. Ukuran file kecil

Skrip dari JavaScript memiliki ukuran yang kecil sehingga ketika web yang memiliki JavaScript ditampilkan di *browser*, maka akses tampilannya akan lebih cepat dibandingkan ketika browser membuka suatu web yang memiliki skrip Java. Hal ini juga sangat berkaitan dengan daya kerja *server*. Semakin kecil space suatu web yang disimpan dalam suatu server, maka daya kerja server ketika di browsing oleh *user* di internet tidak akan terlalu berat. Selain itu sifat JavaScript Client Side yang tidak perlu lagi ditangani oleh *server* ketika browser memanggil web dari sebuah *server*.

2. Mudah untuk dipelajari

JavaScript merupakan bahasa semi pemrograman yang merupakan gabungan antara bahasa pemrograman Java dengan bahasa kode HTML sehingga disebut

juga bahasa hybrid. Walaupun JavaScript merupakan turunan dari Java, namun JavaScript tidak memiliki aturan serumit Java.

3. Terbuka

JavaScript tidak terikat oleh hardware maupun software tertentu, bahkan sistem operasi seperti Windows maupun Unix. Karena ia bersifat terbuka, maka ia dapat dibuat maupun dibaca di semua jenis komputer.

Sedangkan kekurangan JavaScript diantaranya adalah :

1. Skrip tidak terenkripsi

Karena JavaScript bersifat client side, maka skrip yang kita buat di text editor dantelah dijadikan web di server, ketika user me-request web dari server tersebut, maka sintaks JavaScript akan langsung ditampilkan di browser. *User* bisa melihatnya dan menirunya dari *source*-nya.

2. Kemampuan terbatas

Walaupun JavaScript mampu membuat bentuk web menjadi interaktif dan dinamis, namun JavaScript tidak mampu membuat program aplikasi sendiri seperti Java.

3. Keterbatasan objek

JavaScript tidak mampu membuat kelas-kelas yang bisa menampung objek-objek tambahan seperti Java, karena JavaScript telah memiliki objek yang *built in* pada struktur bahasanya.

II.8 XML

XML (*eXtensible Markup Language*) adalah sebuah bahasa *mark-up* yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan maksud atau tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data seperti pada HTML.

XML dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML (*Standard Generalized Markup Language*) yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan markup language yang tidak kalah hebatnya dengan SGML.

XML bersifat mudah untuk dibaca dan ditulis baik oleh manusia maupun komputer, maka XML merupakan sebuah format yang dapat digunakan untuk pertukaran data (interchange) antar aplikasi dan platform yang berbeda (*platform independent*). Misalnya dari Windows ke Unix, atau dari PC ke Machintosh bahkan dari internet ke handphone dengan teknologi WAP. Metode deskripsi data XML (*self-describing*) membuatnya menjadi pilihan efektif untuk bisnis ke bisnis, solusi antar jaringan, *e-business*, dan aplikasi terdistribusi. XML juga bersifat dapat diperluas (*extensible*), dapat digunakan pada semua bahasa pemrograman, dan datanya dapat ditransfer dengan mudah melalui protokol standar internet seperti HTTP tanpa dibatasi oleh *firewall*. XML lebih fleksible dibanding HTML dalam hal kemampuannya menyimpan informasi dan data. Pada XML kita bisa menyimpan data baik dalam atribut maupun sebagai isi elemen yang diletakkan diantara tag pembuka dan tag penutup.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan ‘<’ dan diakhiri dengan ‘>’), tag penutup (diawali dengan ‘</ ‘diakhiri ‘>’) dan atribut elemen. Bedanya dalam HTML, tag-tag telah didefinisikan dalam bentuk baku, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Namun perlu diperhatikan bahwa XML bersifat *case sensitive* yang artinya membedakan inputan antara huruf besar dan huruf kecil.

Contoh XML sederhana :

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Customer>
```

```
<name>Nila Wulandari</name>
<street>Dago</street>
<city>Bandung</city>
<state>JABAR</state>
<zip>40135</zip>
</Customer>
```

Penjelasan : <Customer> sebagai root, 5 baris dibawahnya (name, street, city, state,zip) merupakan elemen *child*.

Dokumen XML dapat digunakan untuk berbagai macam tujuan, seperti:

1. Sebagai penyimpan data (*database*) yang mudah dibaca oleh *user* karena disimpan dalam bentuk teks.
2. Standard transfer data, dapat digunakan untuk pengiriman data transaksi antar perusahaan, atau mentransfer data dari DBMS yang berbeda (mis: Oracle ke SQL Server).
3. Sebagai acuan membuat bahasa baru, seperti WML (Wireless Markup Language) yang digunakan pada mobile device dengan protokol WAP, atau VoiceXML yang digunakan sebagai bahasa markup untuk pengenalan suara, dialog, aplikasi interaksi respon suara maupun DTMF (seperti aplikasi pengisian pulsa atau call center), dan sebagainya.
4. Sebagai file konfigurasi, di Java dokumen-dokumen XML sering kita jumpai seperti file server.xml dan web.xml yang digunakan Tomcat, atau perintah-perintah query yang disimpan dalam file XML yang dipakai pada framework iBatis atau Hibernate.

II.9 JSON

JSON (JavaScript Object Notation) adalah sebuah standarisasi dokumentasi yang digunakan untuk pertukaran komunikasi data secara *online*. Sama seperti XML (eXtensible Markup Language), keduanyaadalah standar yg saat ini banyak digunakan . Standarisasi ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 , Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman

apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

Kelebihan JSON dibandingkan XML adalah, JSON mendukung penggunaan beberapa jenis tipe data,mulai dari String, Integer, Boolean, Double, Long dan sebagainya. Selain itu dari sisi pengolahan (*parsing*) jauh lebih cepat dibandingkan XML.

JSON terdiri dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (hash table), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector)*, daftar (*list*), atau urutan (*sequence*).

II.9.1 Object pada JSON

Cara lama untuk membuat sebuah object adalah dengan menggunakan kata-kunci *new*.

```
var myJSON = new Object();
```

Metode ini telah diperbaiki. Sekarang untuk membuat object kosong, dapat digunakan dengan hanya mendefinisikan pasangan kurungkurawal.

```
var myJSON = {};
```

Seperti halnya kebanyakan object dasar Javascript, format data sangatlah fleksibel yang ditulis sebagai pasangan nama dan nilai. *Object* memiliki nama yang terkandung dalam property object yang biasanya disebut sebagai nama *object*. *Object* juga memiliki property nilai. Lihat contoh berikut:

```
var myFirstJSON = { "firstName" : "Nila",
                    "lastName"  : "Wulandari",
                    "age"       : 22 };
document.writeln(myFirstJSON.firstName); // Output Nila
document.writeln(myFirstJSON.lastName);  // Output
Wulandari
document.writeln(myFirstJSON.age);  // Output 22
```

Penjelasan : *object* tersebut memiliki tiga *property* atau pasangan nama dan nilai. Nama adalah sebuah string (pada contoh di atas firstName, lastName dan age). Nilai dapat diisi dengan semua jenis *object* Javascript (semua di Javascript adalah object, nilai dapat diisi dengan string, angka, array, fungsi dan Object lainnya). Pada contoh di atas nilai diisi dengan Nila, Wulandari dan 22. Nila dan Wulandari adalah string dan 22 adalah angka. Seperti anda lihat pada contoh, nilai dapat diisi dengan berbagai macam *object*.

JSON juga dapat diakses dengan cara seperti *associative array* :

```
var myFirstJSON = { "firstName" : "Nila",
                    "lastName"  : "Wulandari",
                    "age"       : 22 };
document.writeln(myFirstJSON["firstName"]); // Output Nila
document.writeln(myFirstJSON["lastName"]);  // Output
Wulandari
document.writeln(myFirstJSON["age"]);       // Output 22
```

Jika *object* mengandung *object* lagi di dalamnya, tambahkan titik dan nama di belakangnya.

```
 var myObject = {'color' : 'blue',

                  'animal' : { 'cat' : 'friendly' }
                };
document.writeln(myObject.animal.cat); // output friendly
```

II.10 XMPP

XMPP (Extensible Messaging and Presence Protocol) adalah sebuah standard komunikasi *real-time* berbasis text, suara maupun video dengan teknologi open XML. Di awal sekali orang lebih mengenal Jabber, ya karena XMPP merupakan produk Jabber pada tahun 1999, kemudian diformulasikan oleh XMPP Standard Foundation menjadi standard pada tahun 2004 (XMPP 1.0). XMPP ini bisa digunakan dalam hal-hal berikut: *instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication*, dan *generalized routing of XML data*.

II.10.1 Kelebihan dan Kekurangan XMPP

Kelebihan protokol XMPP :

1. Desentralisasi, arsitektur jaringan XMPP mirip dengan arsitektur jaringan email, setiap orang bisa mengembangkan server pribadi, sehingga tidak ada sebuah server utama.
2. Standar terbuka, tidak dibutuhkan royalti dan tidak terikat dengan vendor manapun.
3. Sejarah dan dukungan, teknologi xmpp telah digunakan sejak 1998. Banyak implementasi telah dibuat untuk *server, client, code, component*, dan *library*, bersama dengan dukungan dari perusahaan besar seperti *Sun Microsystem* dan *Google*.
4. Keamanan, *server* XMPP dapat dibuat terisolasi dari *server* public XMPP (misal : *private company* atau jaringan intranet) dan keamanan yang kuat (Melalui SASL dan TLS) telah dibuat didalam spesifikasi inti XMPP.
5. Flexibilitas, fungsionalitas yang unik dapat dibangun diatas XMPP, untuk menjaga keharmonisan antara dua sistem atau lebih, maka *extensions* umum diatur oleh XMPP software foundation.

Kekurangan protokol XMPP :

1. *Presence Data Overhead*, secara umum, dengan adanya lebih dari 70% trafik antar *server* dari protokol XMPP adalah data kehadiran (misal : data user online) dan hampir 60% nya dikirim secara berlebihan, XMPP memiliki banyak data kehadiran *overhead* yang harus dikirimkan kepada banyak penerima.
2. Skalabilitas, XMPP saat ini mengalami masalah yang pada dasarnya adalah masalah kelebihan yang sama, juga menyangkut *multi-user chat* dan penyedia layanan penerbitan dan berlangganan secara online.
3. Tidak ada data biner, cara XMPP dikodekan sebagai sebuah dokumen XML yang panjang membuatnya tidak mungkin untuk mengirimkan data biner tanpa perubahan. Sehingga, pemindahan *file* menggunakan protokol eksternal, misalnya HTTP.

II.11 Basis Data

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user*.

Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System DBMS*). Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data.

II.11.1 Elemen basis data

1. Entitas

Sekumpulan objek yang terdefinisikan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

2. Atribut

Deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

3. Data Value (Nilai Data)

Data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut.

4. File/Tabel

Kumpulan dari beberapa record.

5. Record/Tuple

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

II.11.2 Kelebihan dan Kekurangan Basis Data

Keleibihan dari basis data:

1. Mengurangi duplikasi data.
2. Meningkatkan integritas data.
3. Memelihara independensi data.
4. Meningkatkan keamanan data.
5. Memelihara konsistensi data.
6. Manipulasi data lebih canggih.
7. Mudah untuk diakses.

Kekurangan dari basis data:

1. Sistem lebih rumit, jadi memerlukan tenaga ahli dalam disain, program dan implementasi.
2. Bila ada akses yang tidak benar, kerusakan dapat terjadi.
3. Karena semua data di tempat terpusat, kerusakan *software* dan *hardware* dapat terjadi.
4. Proses pemeliharaan dapat memakan waktu karena ukurannya yang besar.

II.12 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system* atau DBMS) yang *multi-user* dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah open source software yang dibuat oleh sebuah perusahaan Swedia yaitu MySQL AB. MySQL pertama kali dirintis oleh seorang *progammer database* bernama William Widenius.

MySQL sangat mudah digunakan, reliable dan sangat cepat. MySQL mempunyai fitur-fitur yang sangat mudah dipelajari bagi para penggunanya dan dikembangkan untuk menangani database yang besar dengan waktu yang lebih singkat. Kecepatan, konektivitas dan keamanannya yang lebih baik membuat MySQL sangat dibutuhkan untuk mengakses basis data di internet.

MySQL adalah Relational Database Management System (RDBMS) dan untuk mengakses data dalam basis data relasional, maka bahasa yang digunakan yaitu SQL (*Structured Query Language*). Secara umum, SQL terdiri dari bahasa berikut :

1. *Data Definition Language* (DDL)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya *tabel, view, user*, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

1. SELECT untuk menampilkan data.
2. INSERT untuk menambahkan data baru.
3. UPDATE untuk mengubah data yang sudah ada.
4. DELETE untuk menghapus data.

Kelebihan dari MySQL :

1. Portabilitas.

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak *open source*.

   MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL (*General Public License*)sehingga dapat digunakan secara gratis.

3. *Multi-user*.

   MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *'Performance tuning'*.

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data.

   MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi.

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

7. Keamanan.

   MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan.

   MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. Lokalisasi.

    MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka.

    MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Klien dan Peralatan.

    MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

13. Struktur tabel.

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

II.13 Perangkat Lunak Pendukung

II.13.1 Macromedia dreamweaver

Macromedia Dreamweaver dalah suatu bentuk program editor web yang dibuat oleh macromedia. Melalui program ini seorang programmer web dapat dengan mudah membuat dan mendesain webnya. Selain itu, Macromedia Dreamweaver memiliki dua layar, yaitu bentuk halaman Design untuk mengatur dan merancang halaman muka website, dan halaman Code untuk mengetikkan sintak-sintak bahasa pemrograman yang digunakan.

Dengan kelebihan ini, seorang programmer dapat langsung melihat hasil buatannya tanpa harus di browser. Selain mendukung pembuatan web yang berbasis HTML, juga dapat mendukung program–program web yang lain diantaranya PHP, ASP, Perl, Javascript, dan lain-lain.

II.13.3 XAMPP

XAMPP merupakan salah satu paket installasi Apache, PHP dan MySQL instan yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut. Selain paket installasi instant XAMPP versi 1.6.4 juga memberikan fasiltias pilihan pengunaan PHP4 atau PHP5. Untuk berpindah versi PHP yang ingin digunakan juga sangat mudah dilakukan dengan mengunakan bantuan PHP-Switch yang telah disertakan oleh XAMPP.

Secara konvensional kita perlu menginstall ketiga aplikasi (Apache, PHP dan MySQL) diatas secara terpisah. Untuk mempermudah proses installasi ketiga produk tersebut secara instant kita dapat mengunakan XAMPP, dengan bantuan paket XAMPP ini kita dapat secara mudah menginstall Apache, PHP, MySQL dan beberapa aplikasi tambahan lainnya dengan sekali proses. Pada versi XAMPP terbaru saat Artikel ini ditulis juga telah disertakan fasilitas pilihan untuk berpindah versi PHP4 dan PHP5 yang telah di installasi secara instant sebelumnya, sehingga kita tidak perlu repot memikirkan konfigurasi kedua versi PHP tersebut karena XAMPP yang akan mengaturnya untuk kita.

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan pengunaan Apache web server.

Berikut detail paket installasi yang disertakan pada XAMPP 1.6.4 :

1. Apache 2.2.6
2. MySQL 5.0.45
3. PHP 5.2.4 + PHP 4.4.7 + PEAR
4. PHP-Switch win32 1.0
5. XAMPP Control Version 2.5 from www.nat32.com
6. XAMPP Security 1.0
7. SQLite 2.8.15
8. OpenSSL 0.9.8e
9. phpMyAdmin 2.11.1

10. ADOdb 4.95
11. Mercury Mail Transport System v4.01b
12. FileZilla FTP Server 0.9.23
13. Webalizer 2.01-10