LAMPIRAN A

TAMPILAN ANTARMUKA

LAMPIRAN A

TAMPILAN ANTARMUKA

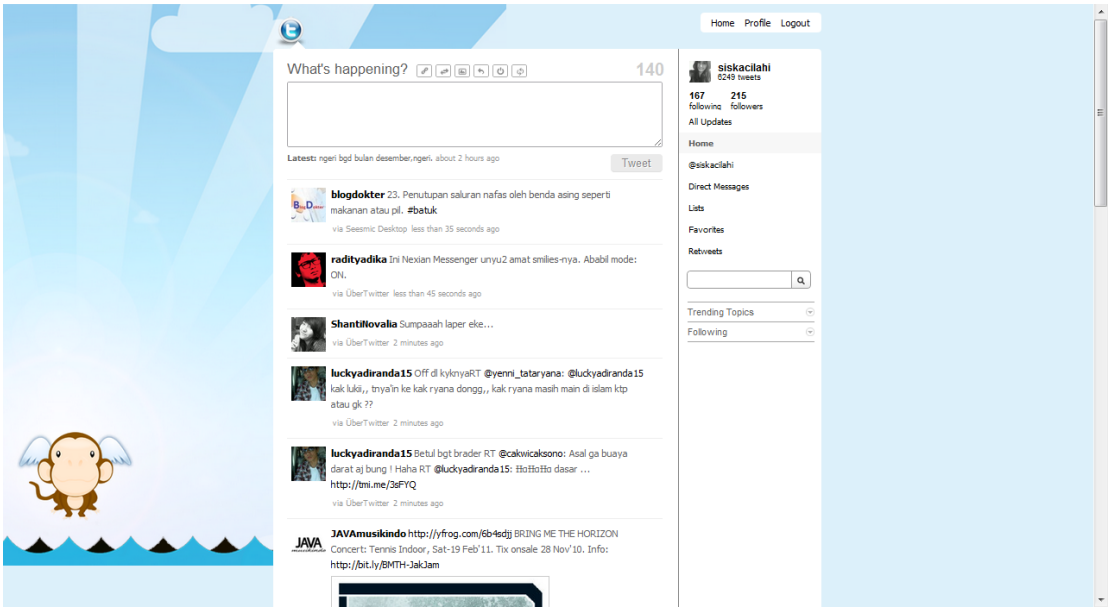
1. Tampilan antarmuka Halaman Depan



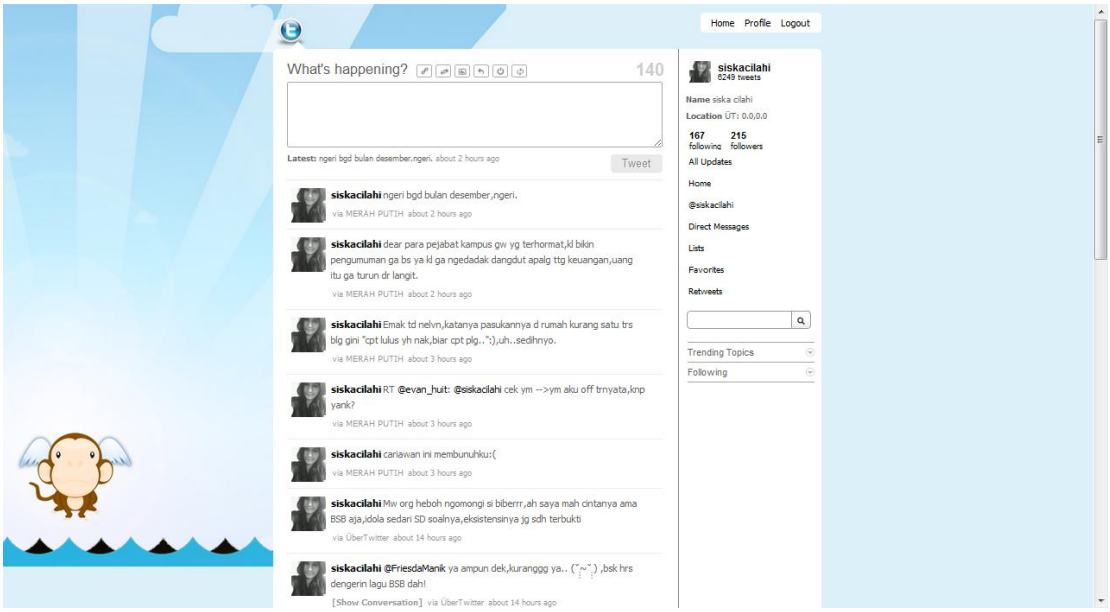
2. Tampilan antarmuka login



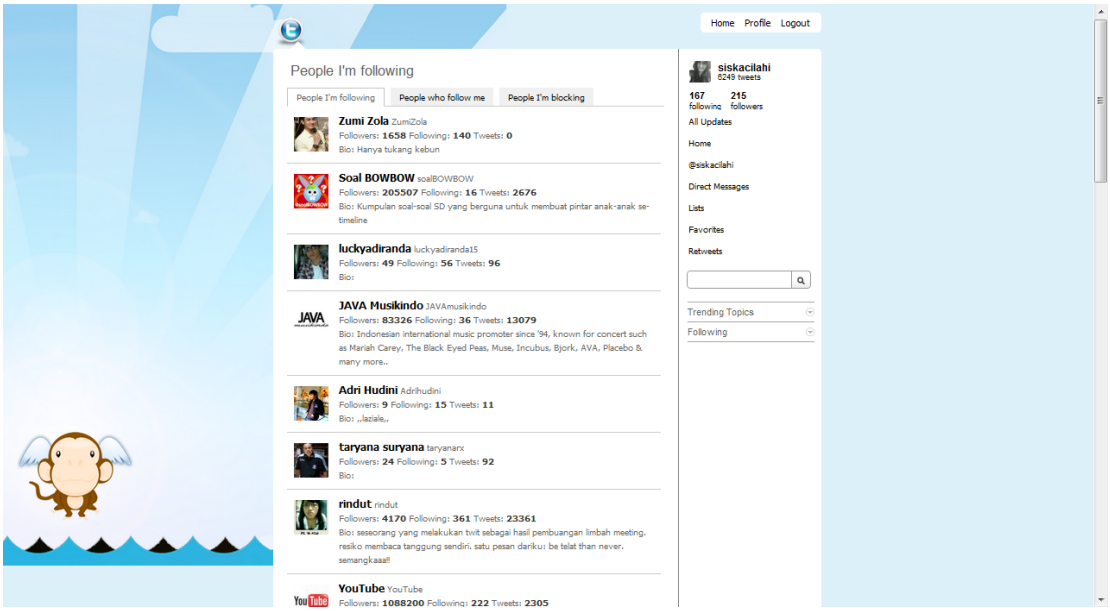
3. Tampilan antarmuka Home



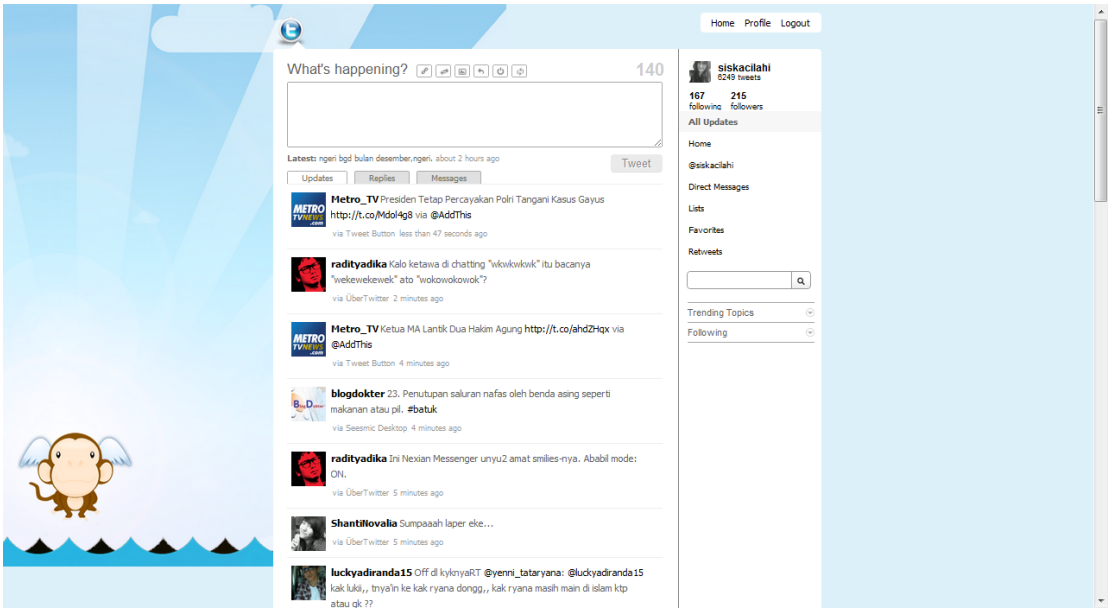
4. Tampilan antarmuka Profile



5. Tampilan antarmuka Follower dan Following



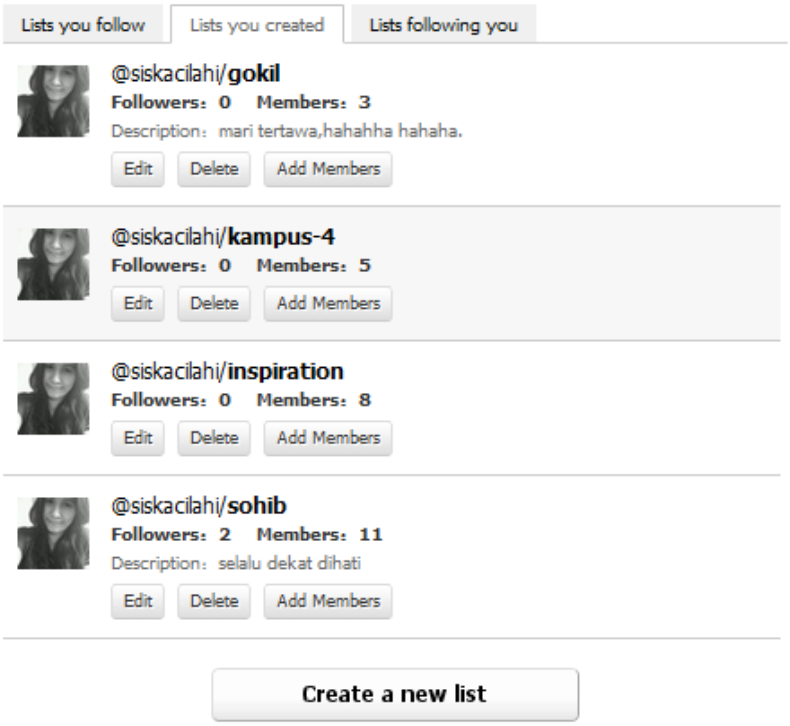
6. Tampilan antarmuka All Updates



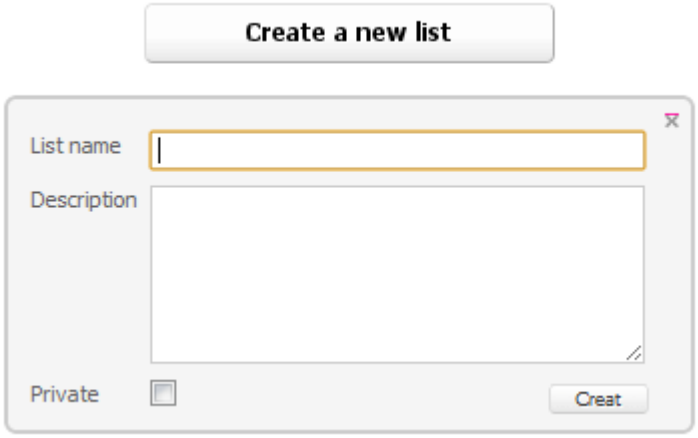
7. Tampilan antarmuka Direct Message



8. Tampilan antarmuka List



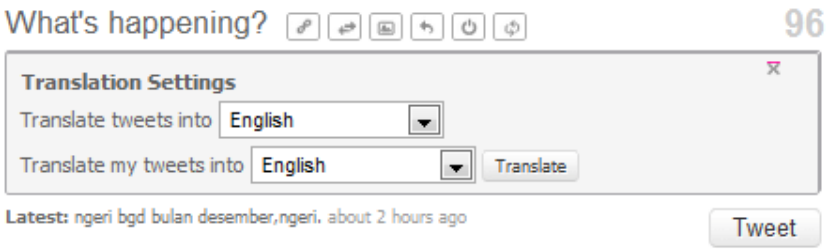
9. Tampilan antarmuka Create List



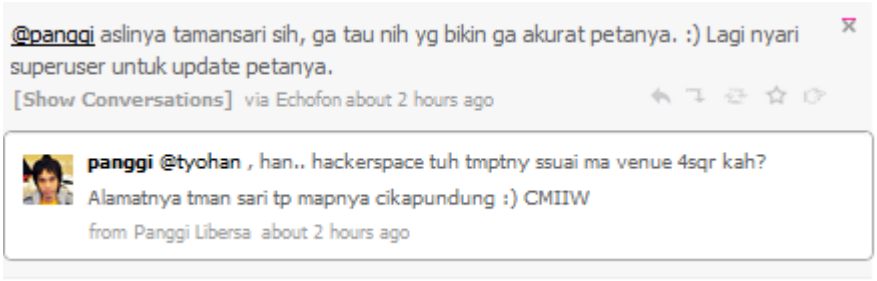
10. Tampilan antarmuka Search



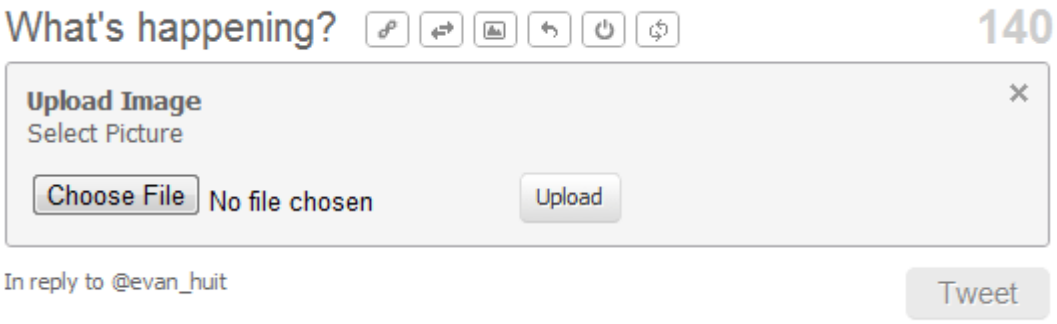
11. Tampilan antarmuka Translation



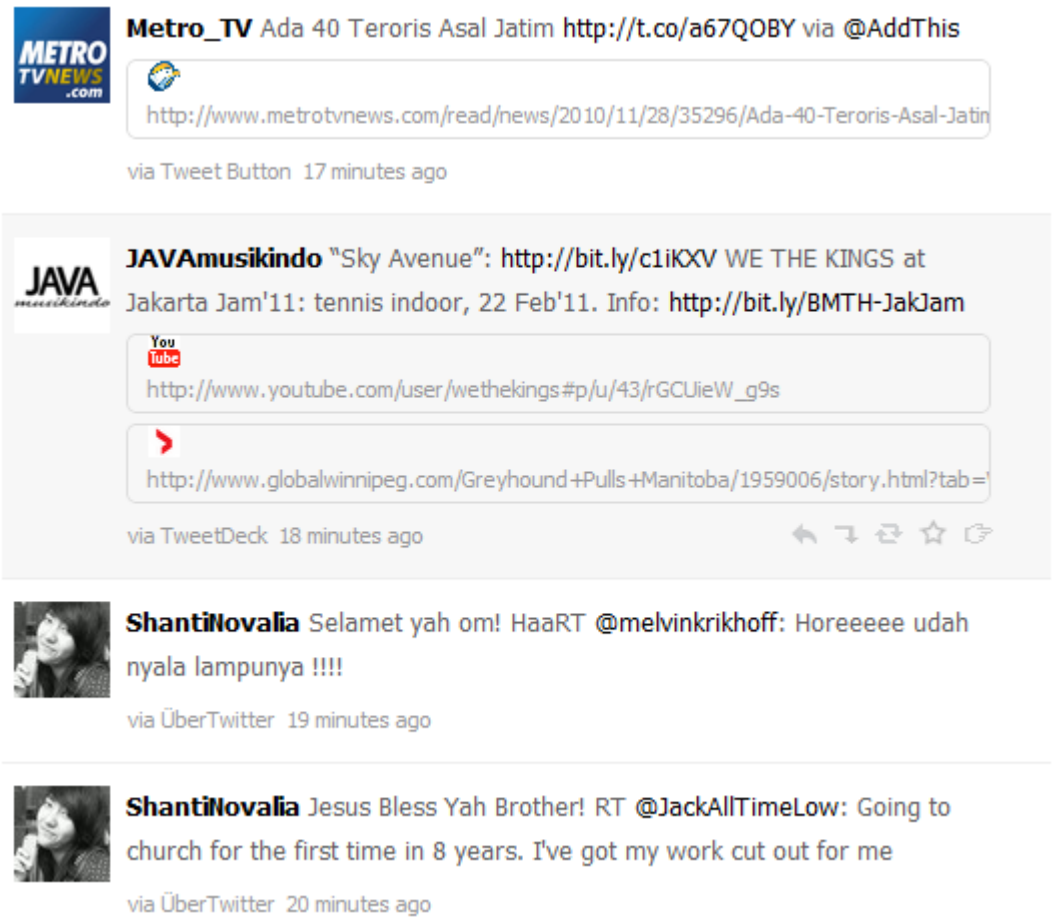
12. Tampilan antarmuka Show Conversation



13. Tampilan antarmuka Upload Image



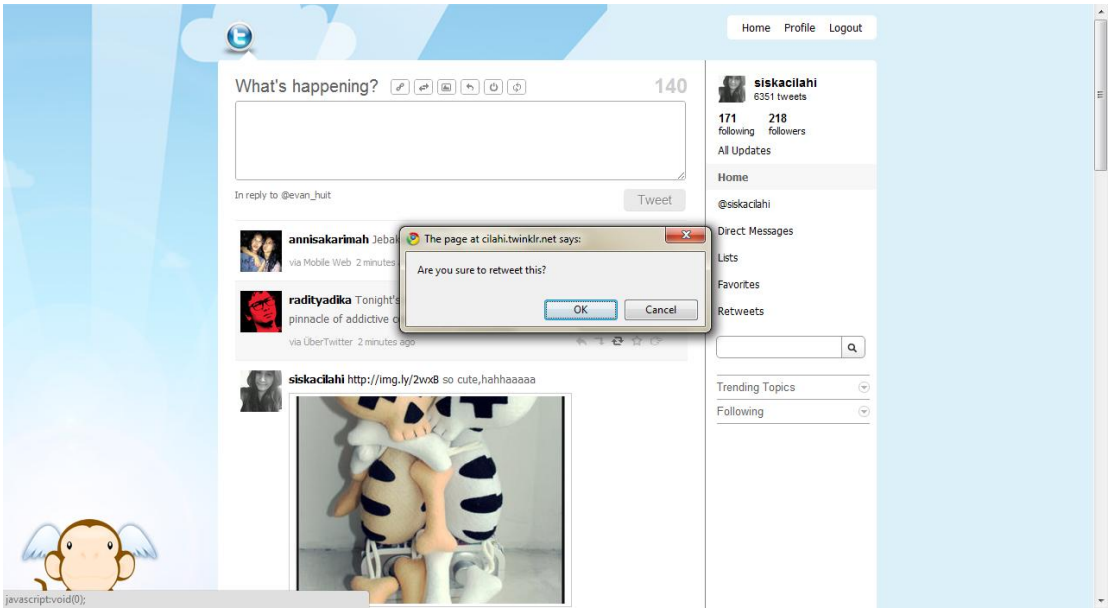
14. Tampilan antarmuka Halaman Timeline



15. Tampilan antarmuka Favorite



16. Tampila antarmuka Retweet



LAMPIRAN B
LISTING PROGRAM

LAMPIRAN B
LISTING PROGRAM

index

```
<?php

error_reporting(0);

	include ('lib/twitese.php');
	$title = "Home";
	include ('inc/header.php');

	if (!loginStatus()) header('location:
login.php');

	$t = getTwitter();

	if (isset($_POST['status']) &&
isset($_POST['in_reply_to'])) {
		if (trim($_POST['status']) == '') {

	setUpdateCookie('empty');
		} else {
			$result = $t-
>update($_POST['status'], $_POST['in_reply_to']);
			if ($result) {

	setUpdateCookie('success');

				$user =
$result->user;
				$time =
time()+3600*24*365;
				if ($user) {

	setcookie('friends_count', $user-
>friends_count, $time, '/');

	setcookie('statuses_count', $user-
>statuses_count, $time, '/');

	setcookie('followers_count', $user-
>followers_count, $time, '/');

	setcookie('imgurl', getAvatar($user-
>profile_image_url), $time, '/');

	setcookie('name', $user->name, $time, '/');
				}
			}
			else {

	setUpdateCookie('error');
			}
		}
		header('location: index.php');
	}
?>
<script type="text/javascript"
src="js/home.js?ver=2010041101"></script>
<div id="statuses" class="column round-left">
<?php

error_reporting(0);  include('inc/sentForm.php'); ?>
```

```
<?php

error_reporting(0);
	$p = 1;
	if (isset($_GET['p'])) {
		$p = (int) $_GET['p'];
		if ($p <= 0) $p = 1;
	}

	$statuses = $t->friendsTimeline($p);
	$retweetes = $t->retweeted_to_me($p);
	if ($statuses === false) {
		header('location: error.php');
	}
	$empty = count($statuses) == 0 ? true:
false;
	if ($empty) {
		echo "<div id=\"empty\">No
tweet to display.</div>";
	} else {
		$output = '<ol class="timeline"
id="allTimeline">';


	include('ajax/timeline_format.php');
		if(count($retweetes) > 0){
			$statuses =
sort_timeline($statuses, $retweetes);
		}
		$MAX_STATUSES = 20;
		$status_count = 0;
		foreach ($statuses as $status) {
			if(++$status_count >=
$MAX_STATUSES){
				break;
			}
			if(isset($status-
>retweeted_status)){
				$output .=
format_retweet($status);
			}else{
				$output .=
format_timeline($status, $t->username);
			}
		}

		$output .= "</ol><div
id=\"pagination\">";

		if ($p >1) $output .= "<a
id=\"more\" class=\"round more\" style=\"float: left;\"
href=\"index.php?p=" . ($p-1) . "\">Back</a>";
		if (!$empty) $output .= "<a
id=\"more\" class=\"round more\" style=\"float:
right;\" href=\"index.php?p=" . ($p+1) . "\">Next</a>";

		$output .= "</div>";

		echo $output;
	}
?>
```

```
</div>
<?php

error_reporting(0);
		include ('inc/sidebar.php');
?>

<?php
error_reporting(0);
		include ('inc/footer.php');
?>
```

Oauth_lib

```
<?php
// vim: foldmethod=marker

/* Generic exception class
 */
class OAuthException extends Exception {
  // pass
}

class OAuthConsumer {
  public $key;
  public $secret;

  function __construct($key, $secret) {
    $this->key = $key;
    $this->secret = $secret;
  }

  function __toString() {
    return "OAuthConsumer[key=$this-
>key,secret=$this->secret]";
  }
}

class OAuthToken {
  // access tokens and request tokens
  public $key;
  public $secret;

  /**
   * key = the token
   * secret = the token secret
   */
  function __construct($key, $secret) {
    $this->key = $key;
    $this->secret = $secret;
  }

  /**
   * generates the basic string serialization of a token
that a server
   * would respond to request_token and access_token
calls with
   */
  function to_string() {
    return "oauth_token=" .
         OAuthUtil::urlencode_rfc3986($this->key) .
         "&oauth_token_secret=" .
         OAuthUtil::urlencode_rfc3986($this->secret);
  }

  function __toString() {
    return $this->to_string();
  }
}

class OAuthSignatureMethod {
  public function check_signature(&$request,
$consumer, $token, $signature) {
    $built = $this->build_signature($request, $consumer,
$token);
    return $built == $signature;
  }
}

class OAuthSignatureMethod_HMAC_SHA1 extends
OAuthSignatureMethod {
  function get_name() {
    return "HMAC-SHA1";
  }

  public function build_signature($request, $consumer,
$token) {
    $base_string = $request-
>get_signature_base_string();
    $request->base_string = $base_string;

    $key_parts = array(
      $consumer->secret,
      ($token) ? $token->secret : ""
    );

    $key_parts =
OAuthUtil::urlencode_rfc3986($key_parts);
    $key = implode('&', $key_parts);

    return base64_encode(hash_hmac('sha1',
$base_string, $key, true));
  }
}

class OAuthSignatureMethod_PLAINTEXT extends
OAuthSignatureMethod {
  public function get_name() {
    return "PLAINTEXT";
  }

  public function build_signature($request, $consumer,
$token) {
    $sig = array(
      OAuthUtil::urlencode_rfc3986($consumer->secret)
    );

    if ($token) {
      array_push($sig,
OAuthUtil::urlencode_rfc3986($token->secret));
    } else {
      array_push($sig, '');
    }

    $raw = implode("&", $sig);
    // for debug purposes
    $request->base_string = $raw;

    return OAuthUtil::urlencode_rfc3986($raw);
  }
}
```

```
class OAuthSignatureMethod_RSA_SHA1 extends
OAuthSignatureMethod {
  public function get_name() {
    return "RSA-SHA1";
  }

  protected function fetch_public_cert(&$request) {
    // not implemented yet, ideas are:
    // (1) do a lookup in a table of trusted certs keyed off
of consumer
    // (2) fetch via http using a url provided by the
requester
    // (3) some sort of specific discovery code based on
request
    //
    // either way should return a string representation of
the certificate
    throw Exception("fetch_public_cert not
implemented");
  }

  protected function fetch_private_cert(&$request) {
    // not implemented yet, ideas are:
    // (1) do a lookup in a table of trusted certs keyed off
of consumer
    //
    // either way should return a string representation of
the certificate
    throw Exception("fetch_private_cert not
implemented");
  }

  public function build_signature(&$request,
$consumer, $token) {
    $base_string = $request-
>get_signature_base_string();
    $request->base_string = $base_string;

    // Fetch the private key cert based on the request
    $cert = $this->fetch_private_cert($request);

    // Pull the private key ID from the certificate
    $privatekeyid = openssl_get_privatekey($cert);

    // Sign using the key
    $ok = openssl_sign($base_string, $signature,
$privatekeyid);

    // Release the key resource
    openssl_free_key($privatekeyid);

    return base64_encode($signature);
  }

  public function check_signature(&$request,
$consumer, $token, $signature) {
    $decoded_sig = base64_decode($signature);

    $base_string = $request-
>get_signature_base_string();

    // Fetch the public key cert based on the request
    $cert = $this->fetch_public_cert($request);

    // Pull the public key ID from the certificate
    $publickeyid = openssl_get_publickey($cert);

    // Check the computed signature against the one
passed in the query
    $ok = openssl_verify($base_string, $decoded_sig,
$publickeyid);

    // Release the key resource
    openssl_free_key($publickeyid);

    return $ok == 1;
  }
}

class OAuthRequest {
  private $parameters;
  private $http_method;
  private $http_url;
  // for debug purposes
  public $base_string;
  public static $version = '1.0a';
  public static $POST_INPUT = 'php://input';

  function __construct($http_method, $http_url,
$parameters=NULL) {
    @$parameters or $parameters = array();
    $this->parameters = $parameters;
    $this->http_method = $http_method;
    $this->http_url = $http_url;
  }


  /**
   * attempt to build up a request from what was
passed to the server
   */
  public static function
from_request($http_method=NULL, $http_url=NULL,
$parameters=NULL) {
    $scheme = (!isset($_SERVER['HTTPS']) ||
$_SERVER['HTTPS'] != "on")
              ? 'http'
              : 'https';
$_SERVER['REQUEST_METHOD'];

    // It's a POST request of the proper content-type, so
parse POST
    // parameters and add those overriding any
duplicates from GET
    if ($http_method == "POST"
        && @strstr($request_headers["Content-Type"],
                   "application/x-www-form-urlencoded")
        ) {
      $post_data = OAuthUtil::parse_parameters(
        file_get_contents(self::$POST_INPUT)
      );
      $parameters = array_merge($parameters,
$post_data);
    }

    // We have a Authorization-header with OAuth
data. Parse the header
    // and add those overriding any duplicates from
GET or POST
```

```
    if (@substr($request_headers['Authorization'], 0, 6)
== "OAuth ") {
      $header_parameters = OAuthUtil::split_header(
        $request_headers['Authorization']
      );
      $parameters = array_merge($parameters,
$header_parameters);
    }

  }

  return new OAuthRequest($http_method, $http_url,
$parameters);
 }

 /**
  * pretty much a helper function to set up the request
  */
 public static function
from_consumer_and_token($consumer, $token,
$http_method, $http_url, $parameters=NULL) {
  @$parameters or $parameters = array();
  $defaults = array("oauth_version" =>
OAuthRequest::$version,
                "oauth_nonce" =>
OAuthRequest::generate_nonce(),
                "oauth_timestamp" =>
OAuthRequest::generate_timestamp(),
                "oauth_consumer_key" => $consumer-
>key);
  if ($token)
    $defaults['oauth_token'] = $token->key;

  $parameters = array_merge($defaults,
$parameters);

  return new OAuthRequest($http_method, $http_url,
$parameters);
 }

 public function set_parameter($name, $value,
$allow_duplicates = true) {
  if ($allow_duplicates && isset($this-
>parameters[$name])) {
    // We have already added parameter(s) with this
name, so add to the list
    if (is_scalar($this->parameters[$name])) {
      // This is the first duplicate, so transform scalar
(string)
      // into an array so we can add the duplicates
      $this->parameters[$name] = array($this-
>parameters[$name]);
    }

    $this->parameters[$name][] = $value;
  } else {
    $this->parameters[$name] = $value;
  }
 }

 public function get_parameter($name) {
  return isset($this->parameters[$name]) ? $this-
>parameters[$name] : null;
 }

 public function get_parameters() {
  return $this->parameters;
 }

 public function unset_parameter($name) {
  unset($this->parameters[$name]);
 }

 /**
  * The request parameters, sorted and concatenated
into a normalized string.
  * @return string
  */
 public function get_signable_parameters() {
  // Grab all parameters
  $params = $this->parameters;

  // Remove oauth_signature if present
  // Ref: Spec: 9.1.1 ("The oauth_signature parameter
MUST be excluded.")
  if (isset($params['oauth_signature'])) {
    unset($params['oauth_signature']);
  }

  return OAuthUtil::build_http_query($params);
 }

 public function get_signature_base_string() {
  $parts = array(
    $this->get_normalized_http_method(),
    $this->get_normalized_http_url(),
    $this->get_signable_parameters()
  );

  $parts = OAuthUtil::urlencode_rfc3986($parts);

  return implode('&', $parts);
 }

 /**
  * just uppercases the http method
  */
 public function get_normalized_http_method() {
  return strtoupper($this->http_method);
 }

 /**
  * parses the url and rebuilds it to be
  * scheme://host/path
  */
 public function get_normalized_http_url() {
  $parts = parse_url($this->http_url);

  $port = @$parts['port'];
  $scheme = $parts['scheme'];
  $host = $parts['host'];
  $path = @$parts['path'];

  $port or $port = ($scheme == 'https') ? '443' : '80';

  if (($scheme == 'https' && $port != '443')
      || ($scheme == 'http' && $port != '80')) {
    $host = "$host:$port";
  }
  return "$scheme://$host$path";
```

```
 }

 /**
  * builds a url usable for a GET request
  */
 public function to_url() {
  $post_data = $this->to_postdata();
  $out = $this->get_normalized_http_url();
  if ($post_data) {
   $out .= '?'.$post_data;
  }
  return $out;
 }

 /**
  * builds the data one would send in a POST request
  */
 public function to_postdata() {
  return OAuthUtil::build_http_query($this-
>parameters);
 }

 /**
  * builds the Authorization: header
  */
 public function to_header() {
  $out ='Authorization: OAuth realm=""';
  $total = array();
  foreach ($this->parameters as $k => $v) {
   if (substr($k, 0, 5) != "oauth") continue;
   if (is_array($v)) {
    throw new OAuthException('Arrays not supported
in headers');
   }
   $out .= ',' .
        OAuthUtil::urlencode_rfc3986($k) .
        '="' .
        OAuthUtil::urlencode_rfc3986($v) .
        '"';
  }
  return $out;
 }

 public function __toString() {
  return $this->to_url();
 }


 public function sign_request($signature_method,
$consumer, $token) {
  $this->set_parameter(
   "oauth_signature_method",
   $signature_method->get_name(),
   false
  );
  $signature = $this-
>build_signature($signature_method, $consumer,
$token);
  $this->set_parameter("oauth_signature", $signature,
false);
 }

 public function build_signature($signature_method,
$consumer, $token) {
  $signature = $signature_method-
>build_signature($this, $consumer, $token);
  return $signature;
 }

 /**
  * util function: current timestamp
  */
 private static function generate_timestamp() {
  return time();
 }

 /**
  * util function: current nonce
  */
 private static function generate_nonce() {
  $mt = microtime();
  $rand = mt_rand();

  return md5($mt . $rand); // md5s look nicer than
numbers
 }
}

class OAuthServer {
 protected $timestamp_threshold = 300; // in seconds,
five minutes
 protected $version = 1.0;           // hi blaine
 protected $signature_methods = array();

 protected $data_store;

 function __construct($data_store) {
  $this->data_store = $data_store;
 }

 public function
add_signature_method($signature_method) {
  $this->signature_methods[$signature_method-
>get_name()] =
   $signature_method;
 }

 // high level functions

 /**
  * process a request_token request
  * returns the request token on success
  */
 public function fetch_request_token(&$request) {
  $this->get_version($request);

  $consumer = $this->get_consumer($request);

  // no token required for the initial token request
  $token = NULL;

  $this->check_signature($request, $consumer,
$token);

  $new_token = $this->data_store-
>new_request_token($consumer);

  return $new_token;
 }
```

```
 /**
  * process an access_token request
  * returns the access token on success
  */
 public function fetch_access_token(&$request) {
   $this->get_version($request);

   $consumer = $this->get_consumer($request);

   // requires authorized request token
   $token = $this->get_token($request, $consumer,
"request");


   $this->check_signature($request, $consumer,
$token);

   $new_token = $this->data_store-
>new_access_token($token, $consumer);

   return $new_token;
 }

 /**
  * verify an api call, checks all the parameters
  */
 public function verify_request(&$request) {
   $this->get_version($request);
   $consumer = $this->get_consumer($request);
   $token = $this->get_token($request, $consumer,
"access");
   $this->check_signature($request, $consumer,
$token);
   return array($consumer, $token);
 }

 // Internals from here
 /**
  * version 1
  */
 private function get_version(&$request) {
   $version = $request-
>get_parameter("oauth_version");
   if (!$version) {
     $version = 1.0;
   }
   if ($version && $version != $this->version) {
     throw new OAuthException("OAuth version
'$version' not supported");
   }
   return $version;
 }

 /**
  * figure out the signature with some defaults
  */
 private function get_signature_method(&$request) {
   $signature_method =
       @$request-
>get_parameter("oauth_signature_method");
   if (!$signature_method) {
     $signature_method = "PLAINTEXT";
   }
   if (!in_array($signature_method,
           array_keys($this->signature_methods))) {
     throw new OAuthException(
       "Signature method '$signature_method' not
supported " .
       "try one of the following: " .
       implode(", ", array_keys($this-
>signature_methods))
     );
   }
   return $this-
>signature_methods[$signature_method];
 }

 /**
  * try to find the consumer for the provided request's
consumer key
  */
 private function get_consumer(&$request) {
   $consumer_key = @$request-
>get_parameter("oauth_consumer_key");
   if (!$consumer_key) {
     throw new OAuthException("Invalid consumer
key");
   }

   $consumer = $this->data_store-
>lookup_consumer($consumer_key);
   if (!$consumer) {
     throw new OAuthException("Invalid consumer");
   }

   return $consumer;
 }

 /**
  * try to find the token for the provided request's
token key
  */
 private function get_token(&$request, $consumer,
$token_type="access") {
   $token_field = @$request-
>get_parameter('oauth_token');
   $token = $this->data_store->lookup_token(
     $consumer, $token_type, $token_field
   );
   if (!$token) {
     throw new OAuthException("Invalid $token_type
token: $token_field");
   }
   return $token;
 }

 /**
  * all-in-one function to check the signature on a
request
  * should guess the signature method appropriately
  */
 private function check_signature(&$request,
$consumer, $token) {
   // this should probably be in a different method
   $timestamp = @$request-
>get_parameter('oauth_timestamp');
   $nonce = @$request-
>get_parameter('oauth_nonce');
```

```
    $this->check_timestamp($timestamp);
    $this->check_nonce($consumer, $token, $nonce,
$timestamp);

    $signature_method = $this-
>get_signature_method($request);

    $signature = $request-
>get_parameter('oauth_signature');
    $valid_sig = $signature_method->check_signature(
      $request,
      $consumer,
      $token,
      $signature
    );

    if (!$valid_sig) {
      throw new OAuthException("Invalid signature");
    }
  }

  /**
   * check that the timestamp is new enough
   */
  private function check_timestamp($timestamp) {
    // verify that timestamp is recentish
    $now = time();
    if ($now - $timestamp > $this-
>timestamp_threshold) {
      throw new OAuthException(
        "Expired timestamp, yours $timestamp, ours
$now"
      );
    }
  }

  /**
   * check that the nonce is not repeated
   */
  private function check_nonce($consumer, $token,
$nonce, $timestamp) {
    // verify that the nonce is uniqueish
    $found = $this->data_store->lookup_nonce(
      $consumer,
      $token,
      $nonce,
      $timestamp
    );
    if ($found) {
      throw new OAuthException("Nonce already used:
$nonce");
    }
  }

}

class OAuthDataStore {
  function lookup_consumer($consumer_key) {
    // implement me
  }

  function lookup_token($consumer, $token_type,
$token) {
    // implement me
  }

  function lookup_nonce($consumer, $token, $nonce,
$timestamp) {
    // implement me
  }

  function new_request_token($consumer) {
    // return a new token attached to this consumer
  }

  function new_access_token($token, $consumer) {
    // return a new access token attached to this
consumer
    // for the user associated with this token if the
request token
    // is authorized
    // should also invalidate the request token
  }

}

class OAuthUtil {
  public static function urlencode_rfc3986($input) {
  if (is_array($input)) {
    return array_map(array('OAuthUtil',
'urlencode_rfc3986'), $input);
  } else if (is_scalar($input)) {
    return str_replace(
      '+',
      ' ',
      str_replace('%7E', '~', rawurlencode($input))
    );
  } else {
    return '';
  }
}


  // This decode function isn't taking into consideration
the above
  // modifications to the encoding process. However,
this method doesn't
  // seem to be used anywhere so leaving it as is.
  public static function urldecode_rfc3986($string) {
    return urldecode($string);
  }

  // Utility function for turning the Authorization:
header into
  // parameters, has to do some unescaping
  // Can filter out any non-oauth parameters if needed
(default behaviour)
  public static function split_header($header,
$only_allow_oauth_parameters = true) {
    $pattern = '/(([-_a-z]*)=("([^"]*)"|([^,]*)),?)/';
    $offset = 0;
    $params = array();
    while (preg_match($pattern, $header, $matches,
PREG_OFFSET_CAPTURE, $offset) > 0) {
      $match = $matches[0];
      $header_name = $matches[2][0];
      $header_content = (isset($matches[5])) ?
$matches[5][0] : $matches[4][0];
      if (preg_match('/^oauth_/', $header_name) ||
!$only_allow_oauth_parameters) {
```

```
      $params[$header_name] =
OAuthUtil::urldecode_rfc3986($header_content);
    }
    $offset = $match[1] + strlen($match[0]);
  }

  if (isset($params['realm'])) {
    unset($params['realm']);
  }

  return $params;
 }

 // helper to try to sort out headers for people who
aren't running apache
 public static function get_headers() {
  if (function_exists('apache_request_headers')) {
   // we need this to get the actual Authorization:
header
   // because apache tends to tell us it doesn't exist
   return apache_request_headers();
  }
  // otherwise we don't have apache and are just going
to have to hope
  // that $_SERVER actually contains what we need
  $out = array();
  foreach ($_SERVER as $key => $value) {
   if (substr($key, 0, 5) == "HTTP_") {
    // this is chaos, basically it is just there to capitalize
the first
    // letter of every word that is not an initial HTTP
and strip HTTP
    // code from przemek
    $key = str_replace(
     " ",
     "-",
     ucwords(strtolower(str_replace("_", " ",
substr($key, 5))))
    );
    $out[$key] = $value;
   }
  }
  return $out;
 }

 // This function takes a input like a=b&a=c&d=e and
returns the parsed
 // parameters like this
 // array('a' => array('b','c'), 'd' => 'e')
 public static function parse_parameters( $input ) {

  // Parameters are sorted by name, using
lexicographical byte value ordering.
  // Ref: Spec: 9.1.1 (1)
  uksort($params, 'strcmp');

  $pairs = array();
  foreach ($params as $parameter => $value) {
   if (is_array($value)) {
    // If two or more parameters share the same
name, they are sorted by their value
    // Ref: Spec: 9.1.1 (1)
    natsort($value);
    foreach ($value as $duplicate_value) {
     $pairs[] = $parameter . '=' . $duplicate_value;
    }
   } else {
    $pairs[] = $parameter . '=' . $value;
   }
  }
  // For each parameter, the name is separated from
the corresponding value by an '=' character (ASCII code
61)
  // Each name-value pair is separated by an '&'
character (ASCII code 38)
  return implode('&', $pairs);
 }
}

?>
```

Twitteroauth

```
<?php
* Twitter OAuth class
 */
class TwitterOAuth {
        /* Contains the last HTTP status code
returned */
        public $http_code;
        /* Contains the last API call */
        public $last_api_call;
        /* Set up the API root URL */
        //public $host =
"https://api.twitter.com/1/";
        public $host = API_URL;
        /* Set timeout default */
        public $timeout = 5;
        /* Set connect timeout */
        public $connecttimeout = 5;
        /* Verify SSL Cert */
        public $ssl_verifypeer = FALSE;
        /* Respons type */
        public $type = 'json';
        /* Decode returne json data */
        public $decode_json = TRUE;
        /* Immediately retry the API call if the
response was not successful. */
        //public $retry = TRUE;
        public $source = 'rabr';

        // user info
        public $username;
        public $screen_name;
        public $user_id;

        /**
         * Set API URLS
         */
        function accessTokenURL()  { return
'https://twitter.com/oauth/access_token'; }
        function authenticateURL() { return
'https://twitter.com/oauth/authenticate'; }
        function authorizeURL()    { return
'https://twitter.com/oauth/authorize'; }
        function requestTokenURL() { return
'https://twitter.com/oauth/request_token'; }

        /**
         * Debug helpers
         */
```

```
	function lastStatusCode() { return $this->http_status; }
	function lastAPICall() { return $this->last_api_call; }

	/**
	 * construct TwitterOAuth object
	 */
	function __construct($consumer_key, $consumer_secret, $oauth_token = NULL, $oauth_token_secret = NULL) {
		$this->sha1_method = new OAuthSignatureMethod_HMAC_SHA1();
		$this->consumer = new OAuthConsumer($consumer_key, $consumer_secret);
		if (!empty($oauth_token) && !empty($oauth_token_secret)) {
			$this->token = new OAuthConsumer($oauth_token, $oauth_token_secret);
			$this->screen_name = $_SESSION['access_token']['screen_name'];
			$this->username = $_SESSION['access_token']['screen_name'];
			$this->user_id = $_SESSION['access_token']['user_id'];
		} else {
			$this->token = NULL;
		}
	}


	/**
	 * Get a request_token from Twitter
	 *
	 * @returns a key/value array containing oauth_token and oauth_token_secret
	 */
	function getRequestToken($oauth_callback = NULL) {
		$parameters = array();
		if (!empty($oauth_callback)) {

	$parameters['oauth_callback'] = $oauth_callback;
		}
		$request = $this->oAuthRequest($this->requestTokenURL(), 'GET', $parameters);
		$token = OAuthUtil::parse_parameters($request);
		$this->token = new OAuthConsumer($token['oauth_token'], $token['oauth_token_secret']);
		return $token;
	}

	/**
	 * Get the authorize URL
	 *
	 * @returns a string
	 */
	function getAuthorizeURL($token, $sign_in_with_twitter = TRUE) {
		if (is_array($token)) {
			$token = $token['oauth_token'];
		}
		if (empty($sign_in_with_twitter)) {
			$url = $this->authorizeURL() . "?oauth_token={$token}";
		} else {
			$url = $this->authenticateURL() . "?oauth_token={$token}";
		}
		return $url;
	}

	/**
	 * Exchange the request token and secret for an access token and
	 * secret, to sign API calls.
	 *
	 * @returns array("oauth_token" => the access token,
	 *		"oauth_token_secret" => the access secret)
	 */
	function getAccessToken($oauth_verifier = FALSE) {
		$parameters = array();
		if (!empty($oauth_verifier)) {

	$parameters['oauth_verifier'] = $oauth_verifier;
		}
		$request = $this->oAuthRequest($this->accessTokenURL(), 'GET', $parameters);
		$token = OAuthUtil::parse_parameters($request);
		$this->token = new OAuthConsumer($token['oauth_token'], $token['oauth_token_secret']);
		return $token;
	}

	/**
	 * GET wrappwer for oAuthRequest.
	 */
	function get($url, $parameters = array()) {
		$response = $this->oAuthRequest($url, 'GET', $parameters);
		if($response == false){
			return false;
		}
		if ($this->type == 'json' && $this->decode_json) {
			return @json_decode($response);
		}elseif($this->type == 'xml' && function_exists('simplexml_load_string')){
			return @simplexml_load_string($response);
		}
		return $response;
	}

	/**
```

```
		 * POST wreapper for oAuthRequest.
		 */
		function post($url, $parameters = array()) {
			$response = $this->oAuthRequest($url, 'POST', $parameters);
			if($response === false){
				return false;
			}
			if ($this->type === 'json' && $this->decode_json) {
				return json_decode($response);
			}elseif($this->type == 'xml' && function_exists('simplexml_load_string')){
				return simplexml_load_string($response);
			}
			return $response;
		}

		/**
		 * DELTE wrapper for oAuthReqeust.
		 */
		function delete($url, $parameters = array()) {
			$response = $this->oAuthRequest($url, 'DELETE', $parameters);
			if($response === false){
				return false;
			}
			if ($this->type === 'json' && $this->decode_json) {
				return json_decode($response);
			}elseif($this->type == 'xml' && function_exists('simplexml_load_string')){
				return simplexml_load_string($response);
			}
			return $response;
		}

		/**
		 * Format and sign an OAuth / API request
		 */
		function oAuthRequest($url, $method, $parameters) {
			if ($url[0] == '/') { //non-twitter.com api shall offer the entire url.
				$url = "{$this->host}{$url}.{$this->type}";
			}
			$request = OAuthRequest::from_consumer_and_token($this->consumer, $this->token, $method, $url, $parameters);
			$request->sign_request($this->sha1_method, $this->consumer, $this->token);
			switch ($method) {
			case 'GET':
				return $this->http($request->to_url(), 'GET');
			default:
				return $this->http($request->get_normalized_http_url(), $method, $request->to_postdata());
			}
		}

		/**
		 * Make an HTTP request
		 *
		 * @return API results
		 */
		function http($url, $method, $postfields = NULL) {
			$ci = curl_init();
			/* Curl settings */
			curl_setopt($ci, CURLOPT_CONNECTTIMEOUT, $this->connecttimeout);
			curl_setopt($ci, CURLOPT_TIMEOUT, $this->timeout);
			curl_setopt($ci, CURLOPT_RETURNTRANSFER, TRUE);
			curl_setopt($ci, CURLOPT_HTTPHEADER, array('Expect:'));
			curl_setopt($ci, CURLOPT_SSL_VERIFYPEER, $this->ssl_verifypeer);

			switch ($method) {
			case 'POST':
				curl_setopt($ci, CURLOPT_POST, TRUE);
				if (!empty($postfields)) {
		curl_setopt($ci, CURLOPT_POSTFIELDS, $postfields);
				}
				break;
			case 'DELETE':
				curl_setopt($ci, CURLOPT_CUSTOMREQUEST, 'DELETE');
				if (!empty($postfields)) {
					$url = "{$url}?{$postfields}";
				}
			}

			curl_setopt($ci, CURLOPT_URL, $url);
			$response = curl_exec($ci);
			$this->http_code = curl_getinfo($ci, CURLINFO_HTTP_CODE);
			$this->last_api_call = $url;
			curl_close ($ci);
			return $response;
		}

		/* ---------- API METHODS ---------- */
		/*			*/
		/* ---------- Block ---------- */
		function blockingIDs(){
			$url = 'blocks/blocking/ids';
			return $this->get($url);
		}
```

```
	function blockingList($page){
		$url = '/blocks/blocking';
		$args = array();
		if($page){
			$args['page'] = $page;
		}
		return $this->get($url, $args);
	}

	function blockUser($id){
		$url = "/blocks/create/$id";
		return $this->post($url);
	}

	function isBlocked($id){
		$url = "/blocks/exists/$id";
		return $this->get($url);
	}

	function unblockUser($id){
		$url = "/blocks/destroy/$id";
		return $this->delete($url);
	}

	/* ---------- Messages ---------- */
	function deleteDirectMessage($id){
		$url =
"/direct_messages/destroy/$id";
		return $this->delete($url);
	}

	function directMessages($page = false,
$since_id = false, $count = null){
		$url = '/direct_messages';
		$args = array();
		if( $since_id )
			$args['since_id'] =
$since_id;
		if( $page )
			$args['page'] = $page;
		return $this->get($url, $args);
	}

	function sendDirectMessage($user, $text){
		$url = '/direct_messages/new';
		$args = array();
		$args['user'] = $user;
		if($text)
			$args['text'] = $text;
		return $this->post($url, $args);
	}

	function sentDirectMessage($page = false,
$since = false, $since_id = false){
		$url = '/direct_messages/sent';
		$args = array();
		if($since)
			$args['since'] =
$since;
		if($since_id)
			$args['since_id'] =
$since_id;
		if($page)
			$args['page'] = $page;
		return $this->get($url, $args);
	}

	/* ---------- List ---------- */
	function addListMember($listid,
$memberid){
		$url = "/$this-
>username/$listid/members";
		$args = array();
		if($memberid){
			$args['id'] =
$memberid;
		}
		return $this->post($url, $args);
	}

	function beAddedLists($username = '',
$cursor = false){
		$url =
"/$username/lists/memberships";
		$args = array();
		if($cursor){
			$args['cursor'] =
$cursor;
		}
		return $this->get($url, $args);
	}

	function createList($name, $description,
$isPortect){
		$url = "/$this->username/lists";
		$args = array();
		if($name){
			$args['name'] =
$name;
		}
		if($description){
			$args['description'] =
$description;
		}
		if($isProtect){
			$args['mode'] =
'private';
		}
		return $this->post($url, $args);
	}

	function createdLists($username = '',
$cursor = false){
		$url = "/$username/lists";
		$args = array();
		if($cursor){
			$args['cursor'] =
$cursor;
		}
		return $this->get($url, $args);
	}

	function deleteList($id){
		$arr = explode('/', $id);
		$url = "/$arr[0]/lists/$arr[1]";
		return $this->delete($url);
	}

	function deleteListMember($id,
$memberid){
```

```
                $arr = explode("/", $id);
                $url =
"$arr[0]/$arr[1]/members";
                $args = array();
                if($memberid){
                        $args['id'] =
$memberid;
                }
                return $this->delete($url, $args);
        }

        function editList($prename, $name,
$description, $isProtect){
                $url = "$this-
>username/lists/$prename";
                $args = array();
                if($name){
                        $args['name'] =
$name;
                }
                if($description){
                        $args['description'] =
$description;
                }
                if($isProtect){
                        $args['mode'] =
"private";
                }
                return $this->post($url, $args);
        }

        function followedLists($username = '',
$cursor = false){
                $url =
"/$username/lists/subscriptions";
                $args = array();
                if($cursor){
                        $args['cursor'] =
$cursor;
                }
                return $this->get($url, $args);
        }

        function followList($id){
                $arr = explode("/", $id);
                $url =
"/$arr[0]/$arr[1]/subscribers";
                return $this->post($url, $args);
        }

        function isFollowedList($id){
                $arr = explode('/', $id);
                $url =
"/$arr[0]/$arr[1]/subscribers/$this->username";
                return $this->get($url);
        }

        function listFollowers($id, $cursor = false){
                $arr = explode('/', $id);
                $url =
"/$arr[0]/$arr[1]/subscribers";
                $args = array();
                if($cursor){
                        $args['cursor'] =
$cursor;
                }
                return $this->get($url, $args);
        }

        function listInfo($id){
                $arr = explode('/', $id);
                $url = "/$arr[0]/lists/$arr[1]";
                return $this->get($url);
        }

        function listMembers($id, $cursor = false){
                $arr = explode("/", $id);
                $url =
"/$arr[0]/$arr[1]/members";
                $args = array();
                if($cursor){
                        $args['cursor'] =
$cursor;
                }
                return $this->get($url, $args);

        }

        function listStatus($id, $page = false,
$since_id = false){
                $arr = explode('/', $id);
                $url =
"/$arr[0]/lists/$arr[1]/statuses";
                $args = array();
                if($page){
                        $args['page'] = $page;
                }
                if($since_id){
                        $args['since_id'] =
$since_id;
                }
                return $this->get($url, $args);
        }

        function unfollowList($id){
                $arr = explode("/", $id);
                $url =
"/$arr[0]/$arr[1]/subscribers";
                return $this->delete($url);
        }

        /* ---------- Friendship ---------- */
        function destroyUser($id){
                $url =
"/friendships/destroy/$id";
                return $this->delete($url);
        }

        function followers($id = false, $page = false,
$count = 30){
                $url = '/statuses/followers';
                $url .= $id ? "/$id" : "";
                if( $id )
                        $args['id'] = $id;
                if( $count )
                        $args['count'] = (int)
$count;
                $args['cursor'] = $page ? $page :
-1;
                return $this->get($url, $args);
```

```
		}

		function followUser($id, $notifications =
false){
			$url = "/friendships/create/$id";
			$args = array();
			if($notifications)
				$args['follow'] = true;
			return $this->post($url, $args);
		}

		function friends($id = false, $page = false,
$count = 30){
			$url = '/statuses/friends';
			$url .= $id ? "/$id" : "";
			$args = array();
			if( $id )
				$args['id'] = $id;
			if( $count )
				$args['count'] = (int)
$count;
			$args['cursor'] = $page ? $page :
-1;
			return $this->get($url, $args);
		}

		function isFriend($user_a, $user_b){
			$url = '/friendships/exists';
			$args = array();
			$args['user_a'] = $user_a;
			$args['user_b'] = $user_b;
			return $this->get($url, $args);
		}

		function
friendship($source_screen_name,$target_screen_nam
e){
			$url = '/friendships/show';
			$args = array();
			$args['source_screen_name'] =
$source_screen_name;
			$args['target_screen_name'] =
$target_screen_name;
			return $this->get($url, $args);
		}

		function relationship($target, $source =
false){
			$url = $url = '/friendships/show';
			$args = array();
			$args['target_screen_name'] =
$target;
			if($source){

		$args['source_screen_name'] = $source;
			}
			return $this->get($url, $args);
		}

		function showUser($id = false, $email =
false, $user_id = false, $screen_name = false){
			$url = '/users/show';
			$args = array();
			if($id)
				$args['id'] = $id;
			elseif($screen_name)
				$args['id'] =
$screen_name;
			else
				$args['id'] = $this-
>user_id;

			return $this->get($url, $args);
		}

		/* ---------- Ratelimit ---------- */
		function ratelimit(){
			$url =
'/account/rate_limit_status';
			return $this->get($url);
		}

		function ratelimit_status(){
			return $this->ratelimit();
		}

		/* ---------- Retweet ---------- */
		function getRetweeters($id, $count = false){
			$url = "/statuses/retweets/$id";
			if($count != false){
				$url .=
"?count=$count";
			}
			return $this->get($url);
		}

		function retweet($id){
			$url = "/statuses/retweet/$id";
			return $this->post($url);
		}

		function retweets($id, $count = 20){
			if($count > 100){
				$count = 100;
			}
			$url = "/statuses/retweets/$id";

	$args = array();
	$args['count'] = $count;
	return $this->get($url,$args);
	  }

		// Returns the 20 most recent retweets
posted by the authenticating user.
		function retweeted_by_me($page = false,
$count = 20, $since_id = false, $max_id = false){
			$url =
'/statuses/retweeted_by_me';
			$args = array();
			if($since_id){
				$args['since_id'] =
$since_id;
			}
			if($max_id){
				$args['max_id'] =
$max_id;
			}
			if($count){
				$args['count'] =
$count;
```

```
                }
                if($page){
                        $args['page'] = $page;
                }
                return $this->get($url, $args);
        }

        // Returns the 20 most recent retweets
posted by the authenticating user's friends.
        function retweeted_to_me($page = false,
$count = false, $since_id = false, $max_id = false){
                $url =
'/statuses/retweeted_to_me';
                $args = array();
                if($since_id){
                        $args['since_id'] =
$since_id;
                }
                if($max_id){
                        $args['max_id'] =
$max_id;
                }
                if($count){
                        $args['count'] =
$count;
                }
                if($page){
                        $args['page'] = $page;
                }
                return $this->get($url, $args);
        }

        function retweets_of_me($page = false,
$count = false, $since_id = false, $max_id = false){
                $url =
'/statuses/retweets_of_me';
                $args = array();
                if($since_id){
                        $args['since_id'] =
$since_id;
                }
                if($max_id){
                        $args['max_id'] =
$max_id;
                }
                if($count){
                        $args['count'] =
$count;
                }
                if($page){
                        $args['page'] = $page;
                }
                return $this->get($url, $args);
        }

        /* ---------- Search ---------- */
        function search($q = false, $page = false,
$rpp = false){
                $searchApiUrl = strpos($this-
>host, "twitter.com") > 0 ? "http://search.twitter.com"
: $this->host;
                $url =
$searchApiUrl.'/search.'.$this->type;
                if(!$q)
                        return false;
                $args = array();
                if($page){
                        $args['page'] = $page;
                }
                if($rpp){
                        $args['rpp'] = $rpp;
                }
                $args['q'] = $q;
                return $this->get($url, $args);
        }

        /* ---------- Spam ---------- */
        function reportSpam($screen_name){
                $url = '/1/report_spam';
                $args = array();
                $args['screen_name'] =
$screen_name;
                return $this->post($url, $args);
        }

        /* ---------- Timeline ---------- */
        function deleteStatus($id){
                $url = $this-
>host.'/statuses/destroy';
                $url .= "/$id.$this->type";
                return $this->delete($url);
        }

        function friendsTimeline($page = false,
$since_id = false, $count = false){
                $url =
'/statuses/friends_timeline';
                $args = array();
                if($page)
                        $args['page'] = $page;
                if($since_id)
                        $args['since_id'] =
$since_id;
                if($count)
                        $args['count'] =
$count;
                return $this->get($url, $args);
        }

        function getFavorites($page =
false,$userid=false){
                if($userid == false){
                        $url = '/favorites';
                }
                else{
                        $url =
'/favorites/'.$userid;
                }

                $args = array();
                if($page)
                        $args['page'] = $page;
                return $this->get($url, $args);
        }

        function makeFavorite($id){
                $url = "/favorites/create/$id";
                return $this->post($url);
        }
```

```
	function publicTimeline($sinceid = false){
		$url =
'/statuses/public_timeline';
		$args = array();
		if($sinceid){
			$args['since_id'] =
$sinceid;
		}
		return $this->get($url, $args);
	}

	function removeFavorite($id){
		$url = "/favorites/destroy/$id";
		return $this->post($url);
	}

	function replies($page = false, $since_id =
false){
		$url = '/statuses/mentions';
		$args = array();
		if($page)
			$args['page'] = (int)
$page;
		if($since_id)
			$args['since_id'] =
$since_id;
		return $this->get($url, $args);
	}

	function showStatus($id){
		$url = "/statuses/show/$id";
		return $this->get($url);
	}

	function update($status, $replying_to =
false){
		try{
			$url =
'/statuses/update';
			$args = array();
			$args['status'] =
$status;
			if($replying_to)

	$args['in_reply_to_status_id'] =
$replying_to;
			return $this-
>post($url, $args);
		}catch(Exception $ex){
			echo $ex->getLine." :
".$ex->getMessage();
		}
	}

	function userTimeline($page = false, $id =
false, $count = false, $since_id = false){
		$url = '/statuses/user_timeline';
		$args = array();
		if($page)
			$args['page'] = $page;
		if($id)
			$args['id'] = $id;
		if($count)
			$args['count'] =
$count;
		if($since_id)
			$args['since_id'] =
$since_id;
		$response = $this->get($url,
$args);
		if(isset($response->error))
		{
			if($response->error
=== 'Not authorized')
			{
				return
'protected';
			}
		}
		return $response;
	}

	function trends(){
		$url =
"http://search.twitter.com/trends.$this->type";
		return $this->get($url);
	}

	/* ---------- Misc. ---------- */
	function twitterAvailable(){
		$url = "/help/test";
		if($this->get($url) == 'ok'){
			return true;
		}
		return false;
	}

	function updateProfile($fields = array()){
		$url = '/account/update_profile';
		$args = array();
		foreach( $fields as $pk => $pv ){
			switch( $pk ){
			case 'name' :
				$args[$pk]
= (string) substr( $pv, 0, 20 );
				break;
			case 'email' :
				if(
preg_match( '/\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
Z]{2,4}\b/i', $pv ) )

	$args[$pk] = (string) $pv;
				break;
			case 'url' :
				$args[$pk]
= (string) substr( $pv, 0, 100 );
				break;
			case 'location' :
				$args[$pk]
= (string) substr( $pv, 0, 30 );
				break;
			case 'description' :
				$args[$pk]
= (string) substr( $pv, 0, 160 );
				break;
			default :
				break;
			}
		}
		return $this->post($url, $args);
```

```
		}

		function veverify(){
			$url =
'/account/verify_credentials';
			return $this->get($url);
		}

		/* ---------- twitese method ---------- */
		function rank($page = false, $count = false){
			$url =
TWITESE_API_URL."/rank.$this->type";
			$args = array();
			if($page){
				$args['page'] = $page;
			}
			if($count){
				$args['count'] =
$count;
			}
			return $this->get($url, $args);
		}

		function browse($page = false, $count =
false){
			$url =
TWITESE_API_URL."/browse.$this->type";
			$args = array();
			if($page){
				$args['page'] = $page;
			}
			if($count){
				$args['count'] =
$count;
			}
			return $this->get($url, $args);
		}
}
<?php
	set_time_limit(15);
	if(!isset($_SESSION)){
		session_start();
	}
	include_once('mobile_device_detect.php');

mobile_device_detect(true,true,true,true,true,true,'htt
ps://t.orzdream.com/',false);
	include_once('config.php');
	include_once('utility.php');
	include_once('twitteroauth.php');
	include_once('oauth_lib.php');

	function refreshProfile(){
		$t = getTwitter();
		$user = $t->veverify();
		$time =
$_SERVER['REQUEST_TIME']+3600*24*365;
		setcookie('friends_count', $user-
>friends_count, $time, '/');
		setcookie('statuses_count', $user-
>statuses_count, $time, '/');
		setcookie('followers_count', $user-
>followers_count, $time, '/');
		setcookie('imgurl', getAvatar($user-
>profile_image_url), $time, '/');
		setcookie('name', $user->name, $time, '/');
		setcookie('listed_count', GetListed($t), $time,
'/');
	}

	function saveStyle($myCSS, $fontsize, $bodyBg) {
		$time = $_SERVER['REQUEST_TIME'] +
3600*24*30;
		setcookie('myCSS', $myCSS, $time);
		setcookie('fontsize', $fontsize, $time);
		setcookie('bodyBg', $bodyBg, $time);
	}

	function resetStyle() {
		delCookie('myCSS');
		delCookie('fontsize');
		delCookie('bodyBg');
	}

	function getColor($name, $default) {
		if (getCookie($name)) return
getCookie($name);
		else return $default;
	}

	function shortUrl($url) {
		return $url;
	}

	function setUpdateCookie($value) {
		setcookie('update_status', $value);
	}

	function getUpdateCookie() {
		if ( isset($_COOKIE['update_status']) ) {
			$update_status =
$_COOKIE['update_status'];
			setcookie('update_status', '',
$_SERVER['REQUEST_TIME']-300);
			return $update_status;
		} else {
			return null;
		}
	}

	function formatText($text) {
		//Èç¹û¿ªÆôÁËÄ§ÊõÒý¹Å\" \' ×ª»ØÀ´
		if (get_magic_quotes_gpc()) {
			$text = stripslashes($text);
		}

		//Íí¼ÓurlÁ´½Ó
		$urlReg =
'/(((http|https|ftp):\/\/){1}([[:alnum:]\-
\.])+(\.)(([[:alnum:]]){2,4})?([[:alnum:]\/+=%#&\:\;_\.~\
?\!\-\,]*))/i';
		$text = preg_replace($urlReg, '<a href="\1"
target="_blank" rel="noreferrer">\1</a>', $text);

		//Íí¼Ó@Á´½Ó
		$atReg = '/\B@{1}(([a-zA-Z0-9\_\.\-])+)/i';
		$text = preg_replace($atReg,    '<a
href="user.php?id=\1" target="_blank">\0</a>', $text);

		//Íí¼Ó list Á´½Ó
```

```
            $listReg = '/(\<a[\w+=\:\%\#\&\.~\?\"\'\/\-
]+\>@{1}([a-zA-Z0-9_\.\-]+)<\/a\>([\/a-zA-Z0-9_\.\-
]+))/i';
            $text = preg_replace($listReg,  '<a
href="list.php?id=\2\3" target="_blank">@\2\3</a>',
$text);

            //Íí¼Ó±êÇ©Á´½Ó
            //$tagReg = "/(\#{1}([a-zA-Z0-9_\.\-
]+))([\s]*)/u";
            $tagReg = "/\B(\#{1}([\w]*[a-zA-
Z_]+[\w]*))([\s]*)/u";
            $text = preg_replace($tagReg, '<a
target="_blank"
href="search.php?q=%23\2">\1</a>\3', $text);

            $text = formatTweetID($text);

            return $text;
        }

        function formatTweetID($text){
            $reg = '/(\<a[\w+=\:\%\#\&\.~\?\"\'\/\-\!
]+\>[\S]+<\/a\>)/i';
            preg_match_all($reg, $text, $tmpMatches);
            if(count($tmpMatches) > 0){
                $text = preg_replace($reg, '$_holder_$',
$text);
            }
            preg_match_all('/([\d]{10,})/', $text,
$matches);
            if(count($matches) > 0){
                $matches = array_unique($matches[0]);
                foreach($matches as $match){
                    $text = str_replace($match, '<a
title="We think it\'s a tweet ID, isn\'t it?"
href="status.php?id='.$match.'"
target="_blank">'.$match.'</a>', $text);
                }
                $tmpReg = '/\$_holder_\$/i';
                foreach($tmpMatches[0] as $match){
                    $text = preg_replace($tmpReg,
$match, $text, 1);
                }
            }
            return $text;
        }

        function formatDate($date, $is_raw = false){
            date_default_timezone_set('Asia/Chongqing');
            $differ = $_SERVER['REQUEST_TIME'] -
strtotime($date);

            if ($is_raw) {
                $dateFormated = date('Y-m-d H:i:s',
strtotime($date));
            } else {
                if ($differ < 0) $differ = 0;
                if ($differ < 60) {
                    $dateFormated = ceil($differ) . '
seconds ago';
                } else if ($differ < 3600) {
                    $dateFormated = ceil($differ/60) . '
minutes ago';
                } else if ($differ < 3600*24) {
                    $dateFormated = 'about ' .
ceil($differ/3600) . ' hours ago';
                } else {
                    $dateFormated = date('Y-m-d H:i:s',
strtotime($date));
                }
            }

            return $dateFormated;
        }

        /* ---------- Recover unshorten urls ---------- */
        function recoverShortens($text){
            $patten =
'/(http[s]?\:\/\/[\w]+[\w\.]*\/[\w\/+=%#&\:_\.~\?\!\-
\,]+)/i';
            preg_match_all($patten, $text, $matches);
            $output = '';
            for($i = 0; $i < count($matches[0]); $i += 2){
                if($recovered =
unshortUrl($matches[0][$i])){
                    $split = explode('/', $recovered);
                    $fav_icon =
'http://www.google.com/s2/favicons?domain='.$split[
2];
                    $output .= "<span
class=\"unshorturl\"><img src=\"$fav_icon\"
alt=\"URL\" align=\"absmiddle\"><a
href=\"$recovered\" target=\"_blank\"
rel=\"noreferrer\">$recovered</a></span>";
                }
            }
            return $output;
        }

            function unshortUrl($text) {
                $urlRegs = array();

                $urlRegs[]
='/http:\/\/tinyurl\.com\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/bit\.ly\/([a-
z0-9]*)/i';
                $urlRegs[] ='/http:\/\/j\.mp\/([a-
z0-9]*)/i';
                $urlRegs[] ='/http:\/\/is\.gd\/([a-
z0-9]*)/i';
                $urlRegs[]
='/http:\/\/retwt\.me\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/ff\.im\/-
([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/tr\.im\/([a-
z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/htxt\.it\/([a-z0-9]{4}[a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/yy\.cx\/([a-
z0-9]*)/i';
                $urlRegs[]
='/http:\/\/aa\.cx\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/digg\.com\/([a-z0-9]{6}[a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/goo\.gl\/fb\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/goo\.gl\/([a-z0-9]*)/i';
```

```
                $urlRegs[]
='/http:\/\/orz\.se\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/zi\.mu\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/knb\.im\/([a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/ff\.im\/([a-
z0-9]*)/i';
                $urlRegs[]
='/http:\/\/sinaurl\.cn\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/163\.fm\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/guao\.cc\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/fl5\.me\/([a-z0-9]*)/i';
                $urlRegs[]
='/http:\/\/9911\.ms\/([a-z0-9]*)/i';
                $urlRegs[] ='/http:\/\/t\.co\/([a-
z0-9]*)/i';
        $urlRegs[] ='/http:\/\/post\.ly\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/short\.ie\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/s8\.hk\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/pendek\.in\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/siteo\.us\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/flic\.kr\/p\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/youtu\.be\/([A-Za-z0-9_?-]*)/i';
                $urlRegs[] =
'/http:\/\/fb\.me\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/nyti\.ms\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/kask\.us\/([a-z0-9]*)/i';
                $urlRegs[] = '/http:\/\/tl\.gd\/([a-
z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/4sq\.com\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/u\.nu\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/om\.ly\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/myloc\.me\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/tumblr\.com\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/on\.ted\.com\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/awe\.sm\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/wp\.me\/([a-z0-9_-]*)/i';
                $urlRegs[] =
'/http:\/\/slidesha\.re\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/ow\.ly\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/lgno\.me\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/crks\.me\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/icio\.us\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/fwd4\.me\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/tnw\.to\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/qhu\.bs\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/blinged\.it\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/cli\.gs\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/lnk\.ms\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/snurl\.com\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/su\.pr\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/twt\.tl\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/twurl\.cc\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/y\.ahoo\.it\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] = '/http:\/\/to\.\/([a-
z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/ping\.fm\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/cot\.ag\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/zd\.net\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/amzn\.to\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/twt\.tl\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/huff\.to\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/pep\.si\/([a-z0-9]*)/i';
                $urlRegs[] = '/http:\/\/n\.pr\/([a-
z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/twurl\.nl\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/short\.to\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/gowal\.la\/s\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/tcrn\.ch\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/idek\.net\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/ericri\.es\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/enva\.to\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/mypict\.me\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/mzl\.la\/([a-z0-9]{5}[a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/dld\.bz\/([a-z0-9]*)/i';
                $urlRegs[] =
'/http:\/\/snipr\.com\/([a-z0-9]*)/i';
        $urlRegs[] ='/http:\/\/twkl\.in\/([a-z0-9]*)/i';
```

```
            $urlRegs[] ='/http:\/\/scr\.bi\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/yhoo\.it\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/fdly\.me\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/on\.cnn\.com\/([a-z0-
9]*)/i';
            $urlRegs[] ='/http:\/\/ff\.im\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/de\.tk\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/nku\.me\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/dlvr\.it\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/nxy\.in\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/ht\.ly\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/kck\.st\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/tgr\.ph\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/ln\-s\.net\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/rww\.to\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/mdtm\.pl\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/kompasiana\.com\/([a-
z0-9]*)/i';
            $urlRegs[] ='/http:\/\/t\.co\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/phpc\.in\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/arst\.ch\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/imdb\.to\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/tiny\.me\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/urho\.me\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/go\.tagjag\.com\/([a-z0-
9]*)/i';
            $urlRegs[] ='/http:\/\/kompasiana\.com\/([a-
z0-9]*)/i';
            $urlRegs[] ='/http:\/\/cl\.ly\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/miso\.io\/([a-z0-9]*)/i';
            $urlRegs[] ='/http:\/\/go\.usa\.gov\/([a-z0-
9]*)/i';
            $urlRegs[] ='/http:\/\/go\.nasa\.gov\/([a-z0-
9]*)/i';
            $urlRegs[] ='/http:\/\/econ\.st\/([a-z0-9]*)/i';

            foreach($urlRegs as $urlReg) {
                if(preg_match($urlReg,$text,$match)){
                    $request =
'http://api.unshort.me/?r=' . $match[0];
                    $obj = objectifyXml(processCurl(
$request ));
                    if (isset($obj->resolvedURL)) return
$obj->resolvedURL;
                }
            }
            return false;
        }

        function processCurl($url,$postdata=false)
        {
            $ch = curl_init($url);

            curl_setopt($ch, CURLOPT_RETURNTRANSFER,
true);
            curl_setopt($ch, CURLOPT_VERBOSE, 1);
            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,
0);
            curl_setopt($ch,
CURLOPT_TIMEOUT,5);//added by yegle

            if($postdata !== false) {
                curl_setopt ($ch, CURLOPT_POST, true);
                curl_setopt ($ch, CURLOPT_POSTFIELDS,
$postdata);
            }

            $response = curl_exec($ch);
            $responseInfo=curl_getinfo($ch);
            curl_close($ch);

            if( intval( $responseInfo['http_code'] ) == 200 )
                return $response;
            else
                return false;
        }

        function objectifyXml( $data ){

            if( function_exists('simplexml_load_string') ) {
                $obj = simplexml_load_string( $data );
            }
            if (isset($obj->error) || !$obj) return false;
            else return $obj;

            return false;
        }

        function objectifyJson($data){
            if(function_exists("json_decode")){
                $obj = json_decode($data);
            }
            if(!isset($obj->error) || $obj){
                return $obj;
            }
            return false;
        }


        /* ---------- image upload ---------- */
        function imageUpload($image){
            $t = getTwitter();
            $signingurl =
'https://api.twitter.com/1/account/verify_credentials.j
son';
            $request =
OAuthRequest::from_consumer_and_token($t-
>consumer, $t->token, 'GET', $signingurl, array());
            $request->sign_request($t->sha1_method, $t-
>consumer, $t->token);
            // header
            $header = $request-
>to_header("http://api.twitter.com/");
            $postdata = array
            (
                'media' => "@$image",
            );

            /**** request method ****/
            $url = 'http://img.ly/api/2/upload.xml';
            $ch = curl_init($url);

            if($postdata !== false)
            {
                curl_setopt ($ch, CURLOPT_POST, true);
                curl_setopt ($ch, CURLOPT_POSTFIELDS,
$postdata);
            }
```

```
            curl_setopt($ch, CURLOPT_HTTPHEADER,
array('X-Auth-Service-Provider: '.$signingurl,'X-Verify-
Credentials-'.$header));
            curl_setopt($ch, CURLOPT_RETURNTRANSFER,
true);
            curl_setopt($ch, CURLOPT_USERAGENT,
'embr');
            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,
0);
            curl_setopt($ch, CURLOPT_TIMEOUT,5);
            curl_setopt($ch,
CURLOPT_CONNECTTIMEOUT,30);

            $response = curl_exec($ch);
            $response_info=curl_getinfo($ch);
            curl_close($ch);

            if ($response_info['http_code'] == 200) {
                return objectifyXml($response);
            } else {
                return null;
            }
        }

        function getTwitter() {
            if(loginStatus()){
                $access_token =
$_SESSION['access_token'] ?
$_SESSION['access_token'] : null;
                $oauth_token = $access_token ?
$access_token['oauth_token'] :
$_COOKIE['oauth_token'];
                $oauth_token_secret = $access_token ?
$access_token['oauth_token_secret'] :
$_COOKIE['oauth_token_secret'];
                $oauth = new
TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET,
$access_token['oauth_token'],
$access_token['oauth_token_secret']);
                return $oauth;
            }
            return null;
        }

        function loginStatus() {
            if(isset($_SESSION['login_status'])){
                return $_SESSION['login_status'] ==
'verified' ? true : false;
            }elseif(getEncryptCookie("oauth_token") != ""
&& getEncryptCookie("oauth_token_secret") != "" &&
getEncryptCookie("user_id") != "" &&
getEncryptCookie("twitese_name") != ""){
                $access_token = array("oauth_token" =>
getEncryptCookie("oauth_token"),
"oauth_token_secret" =>
getEncryptCookie("oauth_token_secret"), "user_id" =>
getEncryptCookie("user_id"), "screen_name" =>
getEncryptCookie("twitese_name"));
                $_SESSION['access_token'] =
$access_token;
                $_SESSION['login_status'] = 'verified';
                refreshProfile();
                return true;
            }
            return false;
        }

        function GetListed($t, $cursor = false){
            $lists = $t->beAddedLists($t->username,
$cursor);
            $listed = count($lists->lists);
            if($lists->next_cursor > 1){
                $listed += GetListed($t, $lists-
>next_cursor);
            }
            return  $listed;
        }

        function getAvatar($profileImg){
            if (getcookie('proxify') == 'true') {
                    $profileImg =
str_replace('http://','http://pubinside.appspot.com/',
$profileImg);
            }
            return $profileImg;

            if($source->following == 1 && $target-
>following == 1){
                return 1;
            }
            if($source->following == 1 && $target-
>following != 1){
                return 2;
            }
            if($source->following != 1 && $target-
>following == 1){
                return 3;
            }
            return 9;
        }
?>
```

LAMPIRAN C

HASIL KUESIONER

LAMPIRAN C

HASIL KUESIONER

KUISIONER

Identitas Responden

Nama : Indra
Akun twitter : @inthera_Indie

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

No Responden :

Identitas Responden

Nama : Evan Gabriel Sabattio
Akun twitter : @sabattio

---

1. Apakah aplikasi *twitter client* ini mudah dimengerti?
   a. Ya b. Tidak
2. Apakah penambahan fitur pada aplikasi *twitter client* ini membuat anda bingung ?
   a. Ya b. Tidak
3. Apakah fitur-fitur tambahan tersebut mudah anda gunakan?
   a. Ya b. Tidak
4. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya b. Tidak
5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya b. Tidak
6. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline
7. Apakah secara tampilan aplikasi *twitter client* sudah cukup menarik?
   a. Ya
   b. Tidak
   c. Cukup

Kritik dan saran : memperbanyak karakter

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Andreas
Akun twitter : @andreasHotman

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Rahmad Alfian
Akun twitter : @Rafi_ian

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
    a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
    a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
    a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Putri nanda sari
Akun twitter : @awesomemput

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Rebi
Akun twitter : @imakubieee

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : adri
Akun twitter : @adrihudini

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Firmansyah
Akun twitter : @ Firmansyah

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam c. 3-4 jam
   b. 1-2jam d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
    a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
    a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
    a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Karissa Fiddia
Akun twitter : @karissafiddia

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
    a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
    a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
    a. Ya b. Tidak

Kritik dan Saran :

-

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Panggi Liberta
Akun twitter : @Panggi

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Buat Petunjuk untuk menuju ke ~~bot~~ login.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Anggie
Akun twitter : @gepratiwi

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Tambahin fitur lokasi dong biar kayak foursquare.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Ageng
Akun twitter : @trisyaardiyant

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
    a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
    a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
    a. Ya b. Tidak

Kritik dan Saran :

Udah bagus kok ^_^ !

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Tezar Irvandi
Akun twitter : @Tezarirvandi

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   ~~a~~ Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya ~~b~~. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   ~~a~~ Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   ~~a~~ Ya b. Tidak

Kritik dan Saran :
Bikin versi mobile biar bisa diakses dari banyak tools, dan bisa digunakan dimana saja.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Shanty Novalia
Akun twitter : @shantinovalia

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Tambahan Fitur Setting biar gak bolak balik ke twitter.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Frisda Rotua Remondang
Akun twitter : @Frisdamanik

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Gimana kalo bisa update status ke FB, pasti lbh multi fungsi.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Ayu Dwi Aprilia
Akun twitter : @AyuDwiAprilia

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Udah keren, cuma tampilannya lebih rame dong !!

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Karina Felicia
Akun twitter : @KarinaFelicia

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Tambahan Fitur bisa donot youtube pasti makin keren.

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Evan Gabriel
Akun twitter : @akan_hurt

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   ✓ a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya ✓ b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   ✓ a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   ✓ a. Ya b. Tidak

Kritik dan Saran :

Buat tampilan yang lebih colourfull dan versi mobile

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : Faizah Abdullah
Akun twitter : @nownaizeh

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
    a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
    a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
    a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda

KUISIONER

Identitas Responden

Nama : febriady
Akun twitter : @Febriadie

---

1. Berapa lama waktu yang anda butuhkan untuk mengerti cara kerja aplikasi *twitter client* ini?
   a. <1jam
   b. 1-2jam
   c. 3-4 jam
   d. >5jam

2. Apakah fitur-fitur tambahan pada aplikasi *twitter client* mudah anda gunakan?
   a. Ya
   b. Tidak

3. Apakah fitur-fitur tersebut membuat anda lebih leluasa dan mempermudah dalam berbagi informasi ?
   a. Ya
   b. Tidak

4. Apakah nama (istilah) fitur-fitur yang digunakan mudah dimengerti?
   a. Ya
   b. Tidak

5. Apakah ada fitur tambahan yang anda tidak mengerti penggunaannya ? (jika tidak ada lanjut ke no 7)
   a. Ya
   b. Tidak

6. Apakah konfirmasi dari setiap aksi fitur sudah mudah dimengerti?
   a. Ya
   b. Tidak

7. Fitur apa yang tidak anda mengerti?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet
   f. refresh timeline
   g. sweep timeline

8. Fitur tambahan apa yang sering terjadi eror (tidak sesuai)?
   a. Shorten url
   b. Upload image
   c. translation
   d. show conversation
   e. retweet

f. refresh timeline
g. sweep timeline

9. Apakah secara tampilan aplikasi *twitter client* sudah menarik dan mudah dimengerti?
   a. Ya b. Tidak

10. Setelah mengggunakan aplikasi *twitter* dan kemudian menggunakan aplikasi *twitter client*, apakah menurut anda ada fitur (kecuali fitur tambahan) yang tidak sesuai dengan cara kerja *twitter*?
   a. Ya b. Tidak

11. Setelah keluar dari aplikasi, apakah anda masih dapat mengingat cara penggunaan aplikasi *twitter client* ini ?
   a. Ya b. Tidak

12. Setelah menggunakan *twitter client* ini, apakah anda merasa kegunaan dari *twitter* meningkat dengan adanya fitur-fitur tambahan tersebut?
   a. Ya b. Tidak

Kritik dan Saran :

Terima Kasih Atas Partisipasi Anda