

An aerial photograph of a basketball court with a green center and red outer areas, surrounded by a dense forest. The scene is captured during the 'golden hour' of sunset, with warm, golden light filtering through the trees and casting long shadows across the court and surrounding area. The text 'STUDENT EMAIL' is overlaid in large, white, bold, sans-serif capital letters across the center of the image.

# STUDENT EMAIL

GUI LINKED LIST



An aerial photograph of an outdoor basketball court. The court has a green central area and maroon-colored key areas and three-point arcs. It is surrounded by a dense forest of trees. The scene is captured during the "golden hour" of sunset, with warm, low-angle light filtering through the trees and casting long shadows. The text "BILAL ABDUL | M AKBAR | ZULFIKRI" is overlaid in white, bold, sans-serif capital letters across the middle of the image, with the vertical lines of the text aligning with the court's center line and the three-point arcs.

**BILAL ABDUL | M AKBAR | ZULFIKRI**





# TAHUN 1971

Menjadi awal sejarah email. Roy Tomlinson, seorang mantan MITian yang bekerja pada sistem operasi TENEX untuk BBN teknologi sebagai kontraktor ARPANET, membuat program electronic mail (surat elektronik) atau yang disingkat sebagai email.





# TAHUN 1971

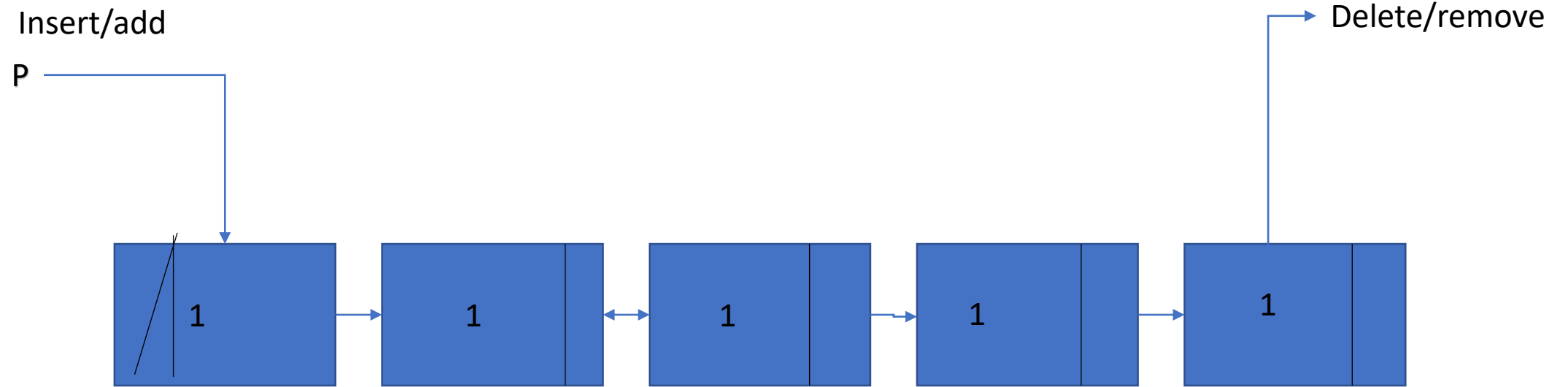
sejarah email. Roy Tomlinson, seorang mantan pekerja pada sistem operasi TENEX untuk BBN sebagai kontraktor ARPANET, membuat program mail (surat elektronik) atau yang disingkat sebagai email.

**Mempermudah untuk menlist email email yang massive di era modern sekarang ini**

# LINKED LIST

Jomblo (single) Linked List





Jomblo (single) Linked List



# EMAIL STUDENT

MacBook Pro

# FITUR

TEST LIST

ADD LIST

DELETE

EDIT

SAVE



# TEST LIST

```
private void addTestData ()  
{  
    nameTextField.setText ("Faiz");  
    idTextField.setText  ("ESQ21");  
    addStudent ();  
  
    nameTextField.setText ("Muhsin");  
    idTextField.setText  ("ESQ31");  
    addStudent ();  
  
    nameTextField.setText ("Bahara");  
    idTextField.setText  ("ESQ41");  
    addStudent ();  
}
```

MacBook Pro



# ADD LIST

```
StudentEmail stud = new StudentEmail (nameTextField.getText(),  
idTextField.getText());
```

```
studentLinkedList.add (stud);
```

```
private void addStudent ()  
{  
    if (isStudentIdInLinkedList (idTextField.getText()) == true)  
    {  
        JOptionPane.showMessageDialog (StudentEmailGUI.this,  
            "Error: student ID is already in the database.");  
    }  
    else  
    {  
        try  
        {  
            StudentEmail stud = new StudentEmail (nameTextField.getText(),  
                idTextField.getText());  
  
            studentLinkedList.add (stud); // tambah student  
  
            displayAll ();  
  
            nameTextField.setText("");  
            idTextField.setText("");  
        }  
        catch (StudentEmailException error)  
        {  
            JOptionPane.showMessageDialog (StudentEmailGUI.this, error.toString ());  
            // myLabel.setText (error.toString ());  
        }  
    }  
}
```



# DELETE LIST

`studentLinkedList.remove (s);`

```
private void deleteStudent ()
{
    if (studentLinkedList.size() == 0)
    {
        JOptionPane.showMessageDialog (StudentEmailGUI.this,
            "Error: Database is empty.");
    }
    else if (isStudentIdInLinkedList (idTextField.getText()) == false)
    {
        JOptionPane.showMessageDialog (StudentEmailGUI.this,
            "Error: student ID is not in the database.");
    }
    else
    {
        for (int s = 0; s < studentLinkedList.size(); s++)
        {
            String currId = studentLinkedList.get (s).getId ();

            if (currId.compareToIgnoreCase (idTextField.getText()) == 0)
            {
                studentLinkedList.remove (s);           // menghapus
            }
        }

        displayAll ();

        nameTextField.setText("");
        idTextField.setText("");
    }
}
```



# EDIT SAVE

```
// Class Instance Data:  
private LinkedList<StudentEmail> studentLinkedList = new LinkedList<StudentEmail> ();  
private int editIndex;
```

```
private void editSaveStudent ()  
{  
    // This code will preserve the changes the user made to the student  
    // they were editing - and save them back into the Linked List.  
  
    studentLinkedList.get (editIndex).setName (nameTextField.getText() );  
    studentLinkedList.get (editIndex).setId   (idTextField.getText() );  
  
    displayAll ();  
  
    nameTextField.setText ("" );  
    idTextField.setText   ("" );  
  
    editSaveButton.setEnabled (false);  
}
```





ESQ BUSINESS SCHOOL

SEKOLAH TINGGI ILMU MANAJEMEN & ILMU KOMPUTER

THANK YOU